Optimal control of a turbulent mixing layer

Sara Delport

Jury:
Em. prof. dr. ir. Y.D. Willems, president
Prof. dr. ir. M. Baelmans, promotor
Prof. dr. ir. J. Meyers, promotor
Prof. dr. M. Diehl
Prof. M.A. Leschziner, PhD, DSc
(Imperial College London)
Prof. dr. ir. E. Van den Bulck
Prof. dr. ir. S. Vandewalle
Prof. dr. ir. G. Winckelmans
(Université catholique de Louvain)

Proefschrift voorgedragen
tot het behalen van de
graad van Doctor in de
ingenieurswetenschappen

December 2010

Kasteelpark Arenberg 1 - bus 2200, B-3001 Leuven (Belgium)

Legal depot number D/2010/7515/122
ISBN number 978-94-6018-283-9

Preface

Vier jaar geleden startte ik met dit onderzoek en vandaag ben ik blij u het resultaat te mogen voorstellen. Het intrigeerde me hoe met actuatoren de stroming beïnvloed kan worden, om zo bijvoorbeeld de wrijvingsweerstand bij boten, auto's of vliegtuigen te verminderen. Er waren direct een heel aantal vragen die bij mij opkwamen zoals welke actuatie de meest optimale is, waar deze best geplaatst wordt, welke vorm, enz. Mijn promotor, professor Tine Baelmans, heeft mij de kans gegeven om, vanuit de expertise van de onderzoeksgroep, te starten met het onderzoek naar optimale actuatie voor turbulente stromingen. Ik ben haar dan ook ten zeerste dankbaar hiervoor.

Dit doctoraat is tot stand gekomen dankzij de hulp van vele mensen. Mijn twee promotoren, professor Tine Baelmans en professor Johan Meyers, waren zeer belangrijk bij dit werk. Ik wil hen bedanken voor de kostbare tijd die zij voor overleg vrijmaakten, overleg dat steeds aanleiding gaf tot nieuwe en verruimde inzichten. Ik ben kunnen starten van de Navier-Stokes code van professor Johan Meyers. Ik heb deze code, en zijn kennis hiervan, echt weten te appreciëren en ik ben dan ook blij dat ik aan deze code de optimalisatie heb kunnen toevoegen.

I also would like to thank the members of the examination committee, for carefully reading the text, their interesting questions, and for their valuable suggestions.

I really enjoyed working at TME for the scientific and non-scientific discussions, the TME-weekends, the lunch breaks, the good atmosphere, ... I want to mention in particular the professors, the fluids engineering group, my current and former office-mates, the secretaries and the technicians. They all made the past four years fabulous.

During my PhD, I spend three months at Imperial College London. Professor Michael Leschziner, Dr. Sylvain Lardeau, Dr. Aimee Morgans and Jeremy Dahan were the key persons during the research visit. Thanks to them, the visit was very interesting and also fun.

Ik wil het Instituut voor de Aanmoediging van Innovatie door Wetenschap en

Technologie in Vlaanderen (IWT-Vlaanderen) bedanken voor het verlenen van een specialisatiebeurs. Het Vlaams Supercomputer Centrum ben ik zeer erkentelijk voor de verleende rekentijd en de technische ondersteuning.

Ten slotte wil ik mijn familie en vrienden bedanken voor al de steun, de hulp en de babbeltjes. In het bijzonder wil ik mijn ouders bedanken voor al de mogelijkheden die zij mij gegeven hebben en Titia voor zo'n fantastische zus te zijn. Mijn ouders en mijn zus zijn al mijn hele leven lang heel belangrijk voor mij en dat zal zo blijven, waar ik ook ben. Bedankt!

Heverlee, december 2010 Sara Delport

Samenvatting

Bij turbulente stromingen kunnen relatief kleine veranderingen sterk groeien en bijvoorbeeld de wrijving in een stroming sterk reduceren of de menging van twee fluida verbeteren. Hierdoor wordt stromingscontrole met behulp van actuatoren mogelijk. Een algemene voorstelling in tijd en ruimte van de verstoringen, veroorzaakt door de actuatoren, bevat onmiddellijk een groot aantal parameters. De optimalisatie van deze parameters laat toe de optimale verstoring te bekomen. Numerieke optimalisatie van stromingscontrole wordt in toenemende mate mogelijk gemaakt dankzij de steeds toenemende rekenkracht van supercomputers. Deze thesis focust op de optimalisatie van open-lus controle, beschreven door een groot aantal parameters, voor turbulente stromingen. De optimalisatie minimaliseert een kostfunctionaal die wordt geformuleerd op basis van de oplossing van een Navier–Stokes simulatie.

Om dit uit te werken en de rekenkracht hanteerbaar te houden, is gekozen voor de optimalisatie van een temporele menglaag. De controleparameters zijn de perturbaties op het initiële gemiddelde snelheidsveld. De perturbaties worden geoptimaliseerd rekening houdend met een lineaire en een niet-lineaire beperking. Omwille van de continuïteitsvergelijking, moeten deze perturbaties divergentie-vrij zijn. Dit vormt een lineaire beperking voor de parameters. De mogelijke perturbaties worden beperkt tot deze met een opgelegd en laag energieniveau. Dit leidt tot een niet-lineaire beperking op de parameters.

Deze thesis levert in essentie drie verschillende bijdragen tot het onderzoeksdomein. De eerste twee bijdragen betreffen het uitwerken van een optimalisatiemethode voor turbulente stromingen die rekening houdt met een niet-lineaire beperking op de parameters. Daarnaast is de impact van controle op menglagen voor lange tijdshorizons onderzocht.

De eerste bijdrage is aan de methode voor het optimaliseren van een groot aantal parameters naar een kostfunctionaal die de oplossing van een Navier-Stokes simulatie vergt. De parameterruimte, gebruikt in dit werk, heeft $\mathcal{O}(10^4 - 10^5)$ dimensies. Dit heeft geleid tot de keuze voor een gradiënt-gebaseerde optimalisatie. Daar de Navier–Stokes vergelijkingen partiële differentiaalvergelijkingen zijn, die

duur zijn om op lossen, is de continue ‘adjoint’-methode gebruikt om de gradiënt efficiënt te berekenen. De keuze van de randvoorwaarden voor de ‘adjoint’ simulatie is besproken voor een algemene kostfunctionaal. Deze bespreking geeft inzicht in de beperkingen op de keuze van de kostfunctionaal, opgelegd door de ‘adjoint’-methode. De convergentie van de gradiënt, berekend met de ‘adjoint’-methode, is aangetoond.

De tweede bijdrage is het vergelijken van twee methodes voor het opleggen van de niet-lineaire beperking op de perturbaties. De geteste methodes, de ‘augmented’ of vermeerderde-Lagrangiaanse methode en de gradiënt-projectie-methode, voldoen aan de Karush–Kuhn–Tucker voorwaarden. De vergelijking toont aan dat de gradiënt-projectie-methode veel robuuster is in combinatie met een ‘adjoint’-gebaseerde berekening van de gradiënt.

Deze methode is toegepast om de menglaag te optimaliseren volgens vijf verschillende kostfunctionalen. De kostfunctionalen zijn gebaseerd op de volgende vijf eigenschappen op de tijdshorizon: de menglaagdikte, de turbulente kinetische energie, de kinetische energie van de gemiddelde snelheid, de totale kinetische energie en de enstrofie. De vergelijking van de verschillende optima toont aan dat de drie eerste kostfunctionalen tot hetzelfde optimum leiden. Dat optimum leidt tot grote 2D vortices die optimaal zijn voor de energieoverdracht van de gemiddelde snelheid naar de turbulente fluctuaties, maar die weinig dissipatie veroorzaken. De twee laatste kostfunctionalen, de kinetische energie en de enstrofie kostfunctionaal, daarentegen leiden tot 3D vortices van verschillende grootte.

Ten slotte wordt ook de impact van controle op menglagen voor lange tijdshorizons onderzocht. Een temporele menglaag heeft de eigenschap gelijkvormig (E.: self-similar) te worden voor lange tijdshorizons. Wanneer menglagen gelijkvormig zijn, is de enstrofie constant in de tijd en evolueert de kinetische energie lineair. Dit zou als gevolg kunnen hebben dat de controle geen impact meer heeft op de stromingsevolutie bij lange tijdshorizons en bijgevolg een optimalisatie van de controle niet meer leidt tot een verbetering van de kostfunctiaal. Desondanks tonen de testen aan dat, tot de langste onderzochte tijdshorizon, de optimalisatie leidt tot een significante daling van de kostfunctiaal. De minimalisatie van de kinetische energie op tijdshorizon T leidt tot een significante versnelling van de transitie naar gelijkvormigheid. De maximalisatie van de enstrofie daarentegen stelt deze transitie uit.

Bijkomende testen met ruis op de optimale perturbaties hebben aangetoond dat de optima robuust zijn. De waarden van de kostfunctionalen verslechteren relatief weinig zelfs wanneer het ruisniveau hoog is.

Abstract

In turbulent flows relatively small perturbations can grow strongly and, for example, reduce drag or enhance mixing of two fluids. This enables flow control with actuators. A general description, in time and space, of the flow perturbations induced by the actuators immediately involves many parameters. Optimisation of these parameters leads to the optimal perturbations. Numerical optimisation of flow control is within reach thanks to the recent advances in computational power of supercomputers. This thesis focuses on optimisation of open-loop control for turbulent flows. The used control is described by a large number of parameters. The optimisation minimises a cost functional that is formulated based on the solution of a Navier-Stokes simulation.

To limit the required computational power, the temporal mixing layer is selected as flow case for the optimisation. The control consists of the perturbations on the initial mean-flow field. The perturbations are optimised subject to a linear and a non-linear constraint. The continuity equation imposes that the perturbations are divergence-free. This is a linear constraint on the parameters. The perturbations are limited to the ones with an imposed, low, energy-level. To this end, the optimisation takes into account a non-linear constraint on the parameters.

This thesis makes three contributions to the domain of research. The first two contributions are on an optimisation method for turbulent flows that takes into account a non linear constraint on the parameters. The third contribution is the investigation of the impact of the control on mixing layers at long time horizons.

The first contribution is to optimisation methods for large parameter spaces, and cost functionals that require the solution of a Navier-Stokes simulation. Because the parameter space used has $\mathcal{O}(10^4-10^5)$ dimensions, gradient-based optimisation is selected. Since the Navier–Stokes equations are computationally expensive partial differential equations, the continuous adjoint method is used to calculate the gradient efficiently. The adjoint boundary conditions are discussed for a general cost functional. This discussion provides insight into the restrictions, imposed by the adjoint method, on the choice of the cost functional. The convergence of the gradient, calculated using the adjoint method, is verified.

The second contribution is the comparison of two methods to impose the non-linear constraint: the augmented Lagrangian method, and the gradient projection method. Both methods satisfy the Karush - Kuhn-Tucker conditions. Comparison shows that the gradient projection method is the more robust technology in combination with the gradient calculated with the adjoint method.

The optimisation method is applied to optimise a mixing layer to five different cost functionals. These are based on the following properties at the time horizon: the momentum thickness, the turbulent kinetic energy, the mean-flow kinetic energy, the total kinetic energy and the enstrophy. The first three cost functionals are shown to lead to the same optimum. That optimum results in large two-dimensional vortex structures with maximum impact on mean momentum, but with low dissipation. The remaining two cost functionals, on the other hand, the kinetic energy and the enstrophy cost functionals, result in optimal solutions that have complex three-dimensional vortex structures at different scales.

Finally, the impact of control on mixing layers at long time horizons is investigated. Temporal mixing layers behave self-similar at long time horizons. In the self-similar state, the enstrophy is constant in time and the kinetic energy evolves linearly. This could imply that the control has no impact on the flow at long time horizons and as such, it would no longer be possible for the optimisation to improve the cost functional. Nevertheless, tests show that, up to the longest time horizon investigated, the optimisation leads to a significant decrease of the cost functional. The minimisation of the kinetic energy at the time horizon T leads to a significant acceleration of the transition to self-similarity. The enstrophy maximisation on the other hand delays the onset of self-similarity.

Additional tests with noise on the optimal perturbations have shown that these are robust. The values of the cost functional increase only slightly even for high levels of noise.

Nomenclature

Lower-case Roman letters

a_i	Coefficient in the Gateaux differential of $\mathscr{J}_{\partial\Omega}$, see eq. (2.43)
b	Coefficient in the Gateaux differential of $\mathscr{J}_{\partial\Omega}$, see eq. (2.43)
c_i	Coefficient in the Gateaux differential of $\mathscr{J}_{\partial\Omega}$, see eq. (2.43)
c_i	Control constraint $c_i(\boldsymbol{\phi}) = 0$
$\boldsymbol{d}$	Projection of the search direction
f	Analytical function
$\boldsymbol{f}$	Forcing term
g	Analytical function
$\boldsymbol{g}$	Conjugate gradient direction
k_c	Cut-off wavenumber
k_{min}	Minimum wavenumber
k_1	Wavenumber in streamwise direction
k_2	Wavenumber in spanwise direction
$\imath$	Imaginary unit
n	$L_1 = n\ \lambda_1$ and $L_2 = n\ \lambda_2$
$\mathbf{n}$	Normal on the boundary $\partial\Omega$, $\mathbf{n} = [n_1, n_2, n_3]$
p	Pressure
$\boldsymbol{q}$	Full state of the flow: $\boldsymbol{q} = [\boldsymbol{u}, p]$, $\boldsymbol{q}(\phi) = \boldsymbol{q}(\mathbf{x}, t, \phi)$
$\boldsymbol{q}^*$	Full adjoint state of the flow: $\boldsymbol{q}^* = [\boldsymbol{u}^*, p^*]$, $\boldsymbol{q}^*(\phi) = \boldsymbol{q}^*(\mathbf{x}, t, \phi)$
r_θ	Momentum thickness growth rate $r_\theta = d\theta/dt/\Delta U$
t	Running time variable
$\boldsymbol{u}$	Velocity field $\boldsymbol{u} = [u_1, u_2, u_3]$, $\boldsymbol{u}(\phi) = \boldsymbol{u}(\mathbf{x}, t, \phi)$
$\hat{\boldsymbol{u}}_{(k_1,k_2)}$	Fourier coefficients of $\boldsymbol{u}$ for wave numbers (k_1, k_2)
$\mathbf{x}$	Coordinate

Upper-case Roman letters

BT	Boundary terms
C	Set of constraints, $C = [c_1, c_2, ..., c_{I_1}]$
CFL	Courant–Friedrichs–Lewy number
$\mathbf{D_z}$	Matrix representing the finite-difference discretisation of $\partial/\partial x_3$ in the normal direction
$\mathscr{D}$	Dissipation of turbulent kinetic energy, see eq. (5.6)
$\mathscr{D}_{\mathscr{E}}$	Dissipation of enstrophy, see eq. (5.8)
E_0	Factor defining the energy of the perturbations, see eq. (3.4)
E_{mode}	Energy per mode (k_1, k_2)
$\mathscr{E}$	Volume averaged enstrophy $\mathscr{E} = (\boldsymbol{\omega}, \boldsymbol{\omega})_T / (2\Omega)$
$\mathbf{F}$	Linear independent parameters, $\boldsymbol{\phi} = \mathbf{A} \cdot \mathbf{F}$
$\mathcal{G}$	Gradient of the cost functional
$\mathcal{H}$	Hessian of the cost functional
J_Ω	Underlying cost functional operator of $\mathscr{J}_\Omega$; functional depending on $\boldsymbol{q}$
$J_{\partial\Omega}$	Underlying cost functional operator of $\mathscr{J}_{\partial\Omega}$; functional depending on $\boldsymbol{q}$
J_T	Underlying cost functional operator of $\mathscr{J}_T$; functional depending on $\boldsymbol{q}$
$\mathscr{J}$	Total cost functional, $\mathscr{J}(\phi) = \mathscr{J}(\boldsymbol{q}(\mathbf{x}, t, \phi))$
$\mathscr{J}_a$	Cost functional value for step length a: $\mathscr{J}(\boldsymbol{q}(\phi^{(k)} + a.\boldsymbol{g}^{(k)}))$, analoguous for $\mathscr{J}_b$, $\mathscr{J}_c$, $\mathscr{J}_u$
$\mathscr{J}_\Omega$	Cost functional contribution from the full space-time domain
$\mathscr{J}_{\partial\Omega}$	Cost functional contribution from the spatial boundaries
$\mathscr{J}_T$	Cost functional contribution from the temporal boundary at time T
$\mathscr{J}_{\scriptscriptstyle ENS}$	Cost functional maximising the enstrophy at T
$\mathscr{J}_{\scriptscriptstyle KE}$	Cost functional minimising the kinetic energy at T
$\mathscr{J}_{\scriptscriptstyle M}$	Cost functional maximising the momentum thickness at T
$\mathscr{J}_{\scriptscriptstyle MFE}$	Cost functional minimising the mean-flow energy at T
$\mathscr{J}_{\scriptscriptstyle TKE}$	Cost functional maximising the turbulent kinetic energy at T
$\mathscr{J}_{final}$	Cost functional value at the end of the optimisation
L	Length of the domain
N	Number of grid points in the spectral directions
$\mathscr{N}$	Navier-Stokes operator $\mathscr{N}\,\boldsymbol{q} = \boldsymbol{f}$
$\mathscr{N}'$	Linearised Navier-Stokes operator $\mathscr{N}'(\boldsymbol{q})\,\delta\boldsymbol{q} = \boldsymbol{f}'$
$\mathscr{N}^*$	Adjoint Navier-Stokes operator $\mathscr{N}^*(\boldsymbol{q})\,\boldsymbol{q}^* = \boldsymbol{f}^*$
$\mathscr{P}$	Production of enstrophy, see eq. (5.5)
$\mathscr{P}_{\mathscr{E}}$	Production of enstrophy, see eq. (5.7)
Re	Reynolds number based on $\delta_\omega/2$ and $\Delta U/2$
Re_θ	Reynolds number based on momentum thickness
Re_λ	Taylor-Reynolds number
T	Time horizon

Lower-case Greek symbols

α	Step length
α	Fourier index corresponding with wavenumber $(\alpha 2\pi/L_1, \beta 2\pi/L_2)$
β	Fourier index corresponding with wavenumber $(\alpha 2\pi/L_1, \beta 2\pi/L_2)$
$\beta^{(k)}$	Factor in the Polak-Ribière conjugate gradient method
δ_ω	Initial vorticity thickness [m]
$\delta \mathscr{J}$	Gateaux differential of the cost functional, $\delta \mathscr{J}(\boldsymbol{q}(\phi), \delta\phi)$
$\delta\phi$	Small change on the control functions
$\delta\boldsymbol{q}$	Sensitivity of $\boldsymbol{q}$ to a change of $\boldsymbol{\phi}$ in the direction $\delta\boldsymbol{\phi}$, $\delta\boldsymbol{q}(\mathbf{x}, t, \phi, \delta\boldsymbol{\phi})$
$\delta\boldsymbol{u}$	Sensitivity of $\boldsymbol{u}$ to a change of $\boldsymbol{\phi}$ in the direction $\delta\boldsymbol{u} = [u_1', u_2', u_3']$, $\delta\boldsymbol{u}(\mathbf{x}, t, \phi, \delta\boldsymbol{\phi})$
ε	Error equal to normalised L_2-norm of the difference between the forward and the adjoint gradient
ε	Volume-averaged dissipation rate
$\eta^{(n)}$	Convergence criterion for the contraint in the augmented Lagrangian method
θ	Momentum thickness
$\boldsymbol{\phi}$	Control functions (perturbations)
$\boldsymbol{\phi}_A$	Control functions used as starting point optimisation for $T = 60$ of group A
$\boldsymbol{\phi}_B$	Control functions used as starting point optimisation for $T = 60$ of group B
$\boldsymbol{\phi}_T$	Control functions (perturbations) optimised to time horizon T
λ	Lagrange multiplier
λ_1	Most unstable wavelength in the streamwise direction
λ_2	Most unstable wavelength in the spanwise direction
λ_2	Second eigenvalue of the symmetric tensor $\mathbf{S^2} + \mathbf{Q^2}$
μ	Quadratic penalty term multiplier
ν	Viscosity
σ	Width of the Gaussian profile
$\boldsymbol{\omega}$	Vorticity, the rotor of the velocity, $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$
$\boldsymbol{\omega}^{(n)}$	Convergence criterion of the augmented Lagrangian method, see eq. (4.4)
$\psi_{(r_1,r_2)}$	Test function $\psi_{(r_1,r_2)}(\mathbf{x}) = \hat{a}(x_3)\exp(\imath(r_1 x_1 + r_2 x_2))$ with $r_m \in (2\pi/L_m)[-(N/2-1), ..., (N/2-1)]$.

Upper-case Greek symbols

Δ	Filter width
ΔE	Total dissipated energy, see eq. (5.3)
$\widetilde{\Delta E}$	Effective total dissipated energy, see eq. (6.2)
ΔU	Velocity difference between the upper and lower layer [m/s]
Δx	Grid spacing
Φ	Discretised ϕ
Ω	Three-dimensional spatial domain
$\partial\Omega$	Spatial boundary of the domain

Subscripts

0	Quantity at time $t = 0$ integrated over the full domain
1	Streamwise direction
2	Spanwise direction
3	Normal direction
T	Quantity at time $t = T$ integrated over the full domain
$\partial\Omega$	Quantity integrated over time $[0, T]$ and the spatial boundaries
Ω	Quantity integrated over time $[0, T]$ and over the full domain

Superscripts

(k)	Outer iteration number in conjugate gradient algorithm
(n)	Number of the sub problem in the augmented Lagrangian method
$\mathbf{a}^*$	Adjoint
$\mathbf{a}'$	Linearised
$\mathbf{a}^\bullet$	Property satisfying the KKT property
$\hat{\mathbf{a}}$	Fourier coefficients of the corresponding variable
$\overline{\mathbf{a}}$	Smoothed
$\mathbf{a}^\mathbf{H}$	Hermitian transpose
$\mathbf{a}^\mathbf{T}$	Transpose

Symbols

(k_1, k_2)	Mode with wavenumber k_1 in streamwise direction and k_2 in spanwise direction
(α, β)	Index number of mode corresponding with wavenumber $(k_1, k_2) = (\alpha\ 2\pi/L_1, \beta\ 2\pi/L_2)$
$(\mathbf{a}, \mathbf{b})$	Inner product over full time and space domain
$(\mathbf{a}, \mathbf{b})_{\partial\boldsymbol{\Omega}}$	Inner product over time $[0, T]$ and the spatial boundaries
$(\mathbf{a}, \mathbf{b})_{\mathbf{t_i}}$	Inner product over the space domain of properties at $t = t_i$
$\langle\mathbf{a}\rangle$	Averaging over plane of homogenity $x_1 - x_2$
$a(t, \boldsymbol{\phi}_T)$	Property evaluated at time t for optimised parameters $\boldsymbol{\phi}_T$

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Thanks to the recent advances in the computational power, direct numerical simulations (DNS) or Large-Eddy simulations (LES) of relatively simple turbulent flow cases are now performed within a few hours on parallel computing systems. This opens up the possibility to optimise turbulent flows based on the information gathered in these simulations.

Currently, the optimisation of turbulent flows is performed with various techniques. The techniques selected depend to a large extent on the number of parameters. For optimisation in relatively small parameter spaces, among others, evolutionary algorithms and artificial neural networks are used (Hilgers and Boersma, 2001; Verstraete, Alsalihi, and Van den Braembussche, 2010). For optimisation in very large parameter spaces on the other hand, gradient-based optimisation with adjoint-based calculation of the gradient is used (Bewley, Moin, and Temam, 2001; Wei and Freund, 2006).

At the onset of this PhD, optimisation of turbulent flows in large parameter spaces was still limited to simple turbulent flow cases with relatively low Reynolds number. Bewley et al. (2001) optimised wall transpiration in a channel to reduce the drag. The wall transpiration was described by $\mathcal{O}(10^7)$ parameters and the Reynolds number in this study was $Re_\tau = 180$. Wei and Freund (2006) optimised a 2D mixing layer to reduce the noise generation. The control was also described by $\mathcal{O}(10^7)$ parameters.

In several of the examples listed above, constraints play an important role (Hilgers and Boersma, 2001; Verstraete et al., 2010). The optimisation of turbulent flows with the adjoint method has not focussed on methods to impose the constraints so far. In Bewley et al. (2001) and Wei and Freund (2006) a penalty term was added to the cost functional. This term penalised the use of much energy for the controls, but it does not impose a constraint.

This work aims at contributing to the field of optimisation in large parameter spaces of turbulent flows, and more in particular the optimisation of active open-loop control for turbulent flows. Some of the current challenges are to optimise three-dimensional flows at fully turbulent Reynolds numbers, to perform optimisation with constraints, and to optimise complexer turbulent flow cases.

The research, on turbulent flow optimisation, in the current work is employing the turbulent temporal mixing layer (see, e.g., Fathali et al., 2008; Vreman, Geurts, and Kuerten, 1997; Rogers and Moser, 1992). The temporal mixing layer can be simulated with a code using the very fast and accurate pseudo-spectral discretisation in two directions, on top of that, the simulation domain of the temporal mixing layer is relatively small, as a consequence the computational cost per simulation is acceptable in the framework of optimisation. This allows us to perform a range of optimisations of a fully turbulent mixing layer with the Taylor-Reynolds number Re_λ up to 40.

In the next sections, the current work is first situated in the state of the art of turbulent flow simulation and optimisation. Subsequently, in section 1.2, the mixing layer flow case is introduced. Finally the aims and objectives are specified, followed by the outline of the text.

1.1 Flow simulation and optimisation

Before the use of flow simulations for design optimisation is discussed in section 1.1.2, an introduction to turbulent flow simulations is given in section 1.1.1.

1.1.1 Turbulent flow simulations

The current section gives some background on turbulent flow simulations. Readers familiar with this topic can immediately go to section 1.1.2.

Turbulent flows are three-dimensional, time-dependent and seemingly random (see, e.g. Pope, 2000; Ferziger and Peric, 2002). They are described by the Navier–Stokes equations, expressing mass conservation and momentum conservation. These partial differential equations have only analytical solutions for a limited number of very simple cases (Pope, 2000). Most cases have to be solved numerically. This is quite challenging since the time scales and spatial scales of turbulent motion range from small to large, and if the Navier–Stokes equations are solved without any modelling, called direct numerical simulation (DNS), all scales have to be simulated. As a result of this requirement, the computational cost of a DNS simulation increases with Re^3 (for more information, see, e.g., Pope, 2000; Ferziger and Peric, 2002) with Re the Reynolds number[1], so DNS is only affordable for moderately high Reynolds numbers (range of scales is narrower) and relatively simple cases (less cells).

There are two common approaches to solve the Navier–Stokes equations that are computationally less demanding than DNS: Large-Eddy simulation (LES) and Reynolds Averaged Navier–Stokes (RANS). Both use a model to reduce the computational cost of resolving all time and length scales. LES models the impact of the small turbulent scales, while it still solves the behaviour of the large scales. This allows to use coarser grids and to reduce the calculation time. RANS only solves for the mean-flow field and uses models for the Reynolds stresses. This also enables the use of even coarser grids. (For more details, see e.g. Pope (2000); Ferziger and Peric (2002)). In the current work, DNS is used.

1.1.2 Use of flow simulations for design optimisation

Flow simulations are now commonly used in industry to evaluate designs and to search for improvements. Interpretation of the simulation results leads to design changes. Turbulent flows however are often difficult to predict, because it is difficult to anticipate the impact of a certain small change. Examples where small modifications to the design seriously changed the flow properties are for example given in Cattafesta et al. (2008) (cavity flow), Jameson, Martinelli, and Pierce (1998) (wing design), Greenblatt and Wygnanski (2000) (e.g. mixing layers, airfoils). The sensitivity of the flow to the design can be used to enhance the performance for industrial applications. In accordance with the terminology used in e.g. Jameson et al. (1998), Greenblatt and Wygnanski (2000), and Cattafesta et al. (2008), we name such a design problem a flow control problem.

The solutions for flow control problems are divided into three categories: passive control, active open-loop control, and active closed-loop control (Choi, Jeon, and Kim, 2008; Cattafesta et al., 2008). Passive control requires no energy input in contrast to active control. Active control is categorised based on the absence, or presence of a feedback system: respectively open-loop and closed-loop control. If active control is open-loop, the resulting flow is not measured and the control signal is predetermined. In case of active closed-loop control, the resulting flow is measured and a feedback signal is provided to the controller. This enables the controller to adapt the control based on the response of the flow system.

---

[1] The Reynolds number is the non-dimensional ratio of a characteristic speed times a characteristic length scale of the flow divided by the viscosity ($Re = U_{ref} \cdot L_{ref}/\nu$). According to the Reynolds-number similarity for incompressible flows (see, e.g. Pope, 2000), this parameter expresses that different incompressible fluids flowing through the same geometry behave the same as long as their ratio of speed over viscosity remains the same. The turbulent flow dealt with in this work is a fluid or gas with low Mach number and relatively high Reynolds number.

Examples of passive control are geometrical modifications, surface modification with roughness elements, addition of a splitter plate, or addition of a small control cylinder (Jameson, 1988; Choi et al., 2008). Active control is among others realised with (synthetic) jets, suction, plasma actuators, wall transpiration (unsteady blowing/suction) and rotating or oscillating bluff bodies. Examples are given in Greenblatt and Wygnanski (2000); Cattafesta et al. (2008); Choi et al. (2008); Bewley et al. (2001).

Active control can be defined by many parameters, which have to be selected (Bewley et al., 2001; Wei and Freund, 2006; Greenblatt and Wygnanski, 2000). In the work of Bewley et al. (2001), for example, the drag in a channel was reduced with wall transpiration. This control was defined by $\mathcal{O}(10^7)$ parameters. In order to find good values for the parameters, the process of searching for parameters needs to be automated. Therefore large parameter space optimisation is desired. The next paragraphs focus on the developments thus far achieved in the field of optimisation in large parameter spaces for flow control.

Jameson was one of the researchers taking the lead in the field of optimisation in large parameter spaces for flow control (Jameson, 1988; Jameson et al., 1998; Jameson and Vassberg, 2001). His research is concentrated in the area of shape optimisation for airplanes. He started using flow simulations based on the Euler equations (so omitting the viscosity effects) and used gradient-based optimisation with calculation of the gradient with the adjoint method (Jameson, 1988). The adjoint method allows one to calculate the gradient at a feasible computational cost, about equal to the cost of one flow simulation, even if the number of parameters is large. Currently the method is applied to Navier–Stokes problems (Jameson et al., 1998; Brezillon and Gauger, 2004; Bewley et al., 2001; Wei and Freund, 2006).

Shape optimisation allows to reduce the drag significantly, while maintaining the lift (Jameson et al., 1998; Brezillon and Gauger, 2004; Kroll et al., 2007; Hazra, Schulz, and Brezillon, 2008). The same optimisation methods using adjoint-based calculation of the gradient can also be applied on active flow-control. This optimisation can exploit the wider range of opportunities offered by active flow control. Active controls can for example inject fluid into the flow, perturb the flow at a certain frequency, etc. Examples of applications of optimisation with the adjoint method to turbulent flow cases are drag reduction in a channel with wall transpiration (Bewley et al., 2001) and noise reduction for a mixing layer with forcing terms (Wei and Freund, 2006).

As mentioned earlier, optimisation is often subject to constraints. Examples of constraints for flow control optimisation are limitation of the energy consumption of the controls, continuity of the flow, and shape constraints in a design problem. The constraints can either relate directly to the controls, and do not depend on the flow solution; or they explicitly depend on the flow solution (see discussion in

Giles and Pierce, 2000; Hinze et al., 2009). Examples of turbulent flow optimisation subject to constraints of the first type are given in Brezillon and Gauger (2004), and Jameson et al. (1998). To penalise the energy consumption of the control Bewley et al. (2001), and Wei and Freund (2006) added a penalisation term to the cost functional, proportional to the energy of the control (Bewley et al., 2001; Wei and Freund, 2006). As discussed by Wei and Freund (2006) the result of the optimisation depends on the chosen weight factor for this term and the energy of the control is not strictly imposed. To really impose a strict energy constraint a constrained optimisation method should be used that complies with the Karush–Kuhn–Tucker (KKT) conditions (see e.g., Luenberger (2003); Nocedal and Wright (2006)).

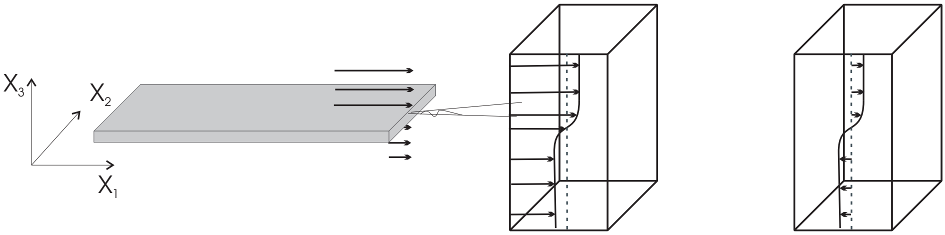
Jameson et al. (1998) refer to the possibility of constraints on the parameters for shape optimisation and suggests to project the gradient in an allowable subspace in which the constraints are satisfied. Brezillon and Gauger (2004) perform shape optimisation with constraints on the lift to drag ratio and also propose to project the gradient. This method is also called the gradient projection method, which is a KKT-compliant method, described in, e.g., Luenberger (2003); Nocedal and Wright (2006); Hinze et al. (2009).

The next section focusses on the current application: turbulent mixing layers (see e.g. Comte, Lesieur, and Lamballais, 1992; Rogers and Moser, 1992, 1994; Moser and Rogers, 1993; Vreman et al., 1997; Balaras, Piomelli, and Wallace, 2001; Fathali et al., 2008). Special attention is paid to the approximative formulation of the mixing layer in the temporal framework, which is used in this work.

1.2 Mixing layer

A mixing layer occurs when two adjacent parallel streams have a different mean velocity. In a wind tunnel this is realised with a splitter plate (see figure 1.1 (a)). The layer above and below the plate have a different mean velocity. Behind the splitter plate the two layers start to interact, perturbations on the mean-flow field trigger instabilities that grow and lead to the development of large coherent structures. The structures start to merge and break up, the thickness of the layer increases and a transition to a fully turbulent state is noticed. This mechanism has been studied both experimentally, and numerically in the previous decades (see e.g., Ref. Brown and Roshko, 1974; Winant and Browand, 1974; Bernal and Roshko, 1986; Metcalfe et al., 1987; Comte et al., 1992; Rogers and Moser, 1992; Moser and Rogers, 1993; Rogers and Moser, 1994; Vreman et al., 1997; Balaras et al., 2001).

Numerical simulations of a mixing layer are performed in the spatial, or the temporal framework (see Figure 1.1). The spatial mixing layer simulates the flow



(a) Spatial (b) Temporal

Figure 1.1: Schematic illustration of a spatial and temporal mixing layer.

in a domain starting from the inlet at the end of the splitter plate (or from further upstream) to the outlet a large extent downstream (see, e.g. Wei and Freund, 2006; Laizet, Lardeau, and Lamballais, 2010). Simulating this large domain requires a large grid, and, as a consequence, much computational time. The temporal framework on the other hand simulates the mixing layer in a comparatively small box (box depicted in figure 1.1) with periodic boundary conditions in the streamwise direction. The initial mean-flow field of the temporal mixing layer has two parallel streams, one in the positive and one in the negative direction (see Figure 1.1 (b)). In case the gradients of the mean velocity in the streamwise direction are vanishingly small compared to the gradients in the normal direction, the evolution of a temporal layer is related to the evolution of a spatial mixing layer for an observer who follows the spatial layer with the average convection speed between the upper and the lower stream (Pope, 2000). However, in case the gradients are non-negligible (when the velocity difference between the streams is significant compared to the average convection speed (Pope, 2000)), the boundary conditions in the temporal framework are not consistent with the spatial boundary conditions as the temporal mixing layer assumes homogeneity in the flow direction.

The main advantage of the temporal framework is its lower computational cost due to the smaller domain, and geometrical simplicity (allowing the use of efficient Fourier discretisation, etc.). Comparison of the results of temporally and spatially evolving mixing layers showed that they are qualitatively and in some cases quantitatively similar (Rogers and Moser, 1994). Rogers, Moser, and Buell (1990) showed that the temporal mixing layer is a good approximation for the spatial mixing layer for velocity ratios as low as 0.2. The self-similarity profiles correspond for the spatial and temporal mixing layer (Pope, 2000; Bell and Mehta, 1990; Rogers and Moser, 1994). Also, the most important physics responsible for transition to turbulence in a mixing layer, e.g., roll-up, pairing, braiding, can be well presented in a temporal layer (for more details, see e.g., Rogers and Moser, 1992; Moser and Rogers, 1993; Vreman et al., 1997). One the other hand the temporal mixing layer grows symmetrically around the middle plane, which does not correspond with the spatial mixing layer, since the latter grows preferentially into the low-speed stream (Pope, 2000).

The initial velocity field of the temporal mixing layer is composed of the mean-flow field and perturbations that trigger the development of instabilities. The streamwise component of the initial mean-flow field has a profile in the normal direction that is often approximated as a hyperbolic-tangent, an error function or a combination of the mean-flow parts of two turbulent boundary layer profiles, while the spanwise and normal components are zero (see, e.g. Pope, 2000; Vreman et al., 1997; Rogers and Moser, 1994). To this profile the perturbations are added. The development of the temporal mixing layer is very sensitive to the perturbations (see, e.g. Moser and Rogers, 1993; Balaras et al., 2001; Fathali et al., 2008). As a consequence, the properties of the mixing layer can be enhanced by altering the perturbations on the initial mean flow field.

For high Reynolds numbers and long time horizons, the temporal mixing layer attains a self-similar state (Wygnanski and Fiedler, 1970; Batt, 1977; Rogers and Moser, 1994; Balaras et al., 2001), characterised by constant profiles (in time) for the mean velocity, the Reynolds stresses, and the production, and the dissipation of turbulent kinetic energy, when normalised appropriately with the mixing-layer thickness and the velocity difference. It is believed that a self-similar region emerges once the inertial effects start dominating the large-scale turbulent motions, accompanied with the emergence of a range of small-scale turbulent eddies displaying high-Reynolds number universal equilibrium behaviour (see, e.g. Dimotakis and Brown, 1976).

1.3 Aims and objectives

This work aims at contributing to the field of optimisation in large parameter spaces of turbulent flows, and more in particular the optimisation of active open-loop control. Some of the current issues in this field are the optimisation of three-dimensional flows at fully turbulent Reynolds numbers, and the optimisation with constraints.

Based on these issues, four objectives are formulated for the current study:

1. implementing an efficient and consistent method to calculate the gradient of a cost functional for a PDE constrained optimisation problem
2. comparing methods that comply with the KKT conditions to take into account a non linear constraint on the control during the optimisation
3. optimising a turbulent flow with different cost functionals to analyse the impact of the choice of the cost functional
4. optimising turbulent flows at a fully turbulent Reynolds number

1.4 Outline

In order to prepare for the optimisation of turbulent flows, chapter 2 elaborates on optimisation techniques, and the methods for constrained optimisation. The application of the selected methods to the mixing layer is discussed in chapter 3, entitled ‘numerical approach’. This chapter also describes the numerical methods used to simulate the mixing layer. In chapter 4 the optimisation is tested and choices with regard to the non linear constraint and the cost functional formulation are made. Five cost functionals are tested. Chapter 5 applies the method developed on the temporal mixing layer with fully turbulent Reynolds number. The effectiveness of low energy controls to improve certain properties at long time horizons is investigated. Chapter 6 tests the robustness of the parameters to the presence of noise. The conclusion is given in chapter 7.

Chapter 2, entitled ‘flow optimisation’, starts with a general definition of a constrained optimisation problem for turbulent flows (section 2.1). In section 2.2 several optimisation techniques are discussed in the context of large parameter space optimisation for flow control and gradient-based descent algorithms are selected. Section 2.3 elaborates on gradient-based descent algorithms. The Polak–Ribière conjugate-gradient method (see, e.g., Ref. Nocedal and Wright (2006)) is selected and combined with the Brent line-search algorithm (Press et al., 1996; Brent, 1973).

The determination of the gradient poses an additional challenge as the number of parameters is high. Therefore we use the adjoint method (see, e.g. Jameson et al., 1998; Bewley et al., 2001), mentioned earlier in section 1.1.2. Section 2.4 elaborates on this method and also discusses the choice of the boundary conditions of the adjoint problem.

In the current manuscript several methods to enforce constraints on the optimisation parameters are tested. The selected methods formally ensure that an optimal point of the constrained optimisation problem is also a KKT point. The methods are the augmented Lagrangian, the gradient projection method, and the parameter elimination method. Section 2.5 elaborates on these techniques.

Chapter 3 starts with the definition of the mixing-layer optimisation problem in section 3.1. The optimisation parameters correspond with the perturbations on the initial mean flow field. A linear and a non-linear constraint are imposed on the parameters. The linear constraint ensures that the perturbations are divergence-free, while the non-linear one fixes their energy-level.

In section 3.2 the computational setup is outlined and the numerical methods used for the mixing-layer flow simulation and adjoint simulation are discussed. Section 3.3 elaborates on the optimisation in the discretised parameter space with special attention to the handling of the constraints on the parameters in the discretised parameter space. The last part of that section also discusses the need to regularise the gradient to ensure that the perturbations remain relevant from a physical point of view.

The numerical approach, outlined in chapter 3, is tested extensively in chapter 4. In section 4.1 the correctness of the gradient calculated with the adjoint method is verified and the convergence properties are discussed (objective 1). Subsequently, in section 4.2, optimisations are performed with both the augmented Lagrangian method and the gradient projection method to select the best method to impose a non linear constraint (objective 2). The gradient projection method is shown to be more robust for this optimisation case than the augmented Lagrangian method.

To analyse the impact of the cost functional, the mixing layer is optimised with respect to five different cost functionals. They are based on commonly monitored properties of mixing layers: the momentum thickness, turbulent kinetic energy, mean-flow kinetic energy, total kinetic energy, and enstrophy. These properties are optimised at the end of the simulation time window, called the time horizon. Optimisation is performed in a three-dimensional turbulent mixing layer with up to $\mathcal{O}(10^4)$ parameters. The total energy of the initial perturbations was kept at 10^{-3} times the energy of the initial mean-flow field. The cost functionals are tested for two different time horizons in section 4.3.2. It is found that the momentum thickness, turbulent kinetic energy and mean-flow kinetic energy cost functionals lead to optimal large-scale structures (with mainly two-dimensional vortex structures), while the total kinetic energy and enstrophy cost functional promote small-scale structures in the flow (objective 3).

The cost functionals leading to small scale structures at the time horizon, the total kinetic energy and enstrophy cost functional, are selected to test the ability to optimise the mixing layer for long time horizons when it attains a fully turbulent Reynolds number (objective 4). These two cost functionals are further employed in chapter 5, and both maximise a dissipation property. The total kinetic energy cost functional maximises the total dissipated energy in the time interval up to the time horizon, and the enstrophy cost functional maximises the dissipation rate at the time horizon. Optimisation is performed in a larger box with up to $\mathcal{O}(10^5)$ parameters, allowing optimisation to longer optimisation time horizons. Two different energy levels (10^{-4} and 10^{-5} times the energy of the initial mean-flow field) for the controls are used. Chapter 5 shows that for the range of time horizons considered in the current study, we find that both energy levels allow us to effectively influence the dissipation properties up to the longest time horizon at which the turbulent Taylor-Reynolds number $Re_\lambda \approx 40$. It was possible to increase both the total dissipated energy and the dissipation rate.

When the mixing-layer evolution is further monitored beyond the optimisation time horizons, it is observed in section 5.3.2 that the solutions optimised to the total dissipated energy evolve very fast to a self-similar state, characterised by a constant growth-rate of the mixing layer, and a constant ratio between production and dissipation of turbulent kinetic energy. The flows that maximise the dissipation rate at the longest time horizons on the other hand delay the onset of self-similarity.

The parameters are optimised in absence of noise on the parameters. In section 6 the impact of background noise on the optimised perturbations is tested. The perturbations optimised in chapter 5 are used, and it is investigated how the total dissipated energy and the dissipation rate is affected when the background noise level is gradually increased in subsequent simulations. It is shown that the optima are robust to the presence of noise.

Chapter 2

Flow optimisation

This chapter introduces the optimisation techniques used in this thesis and motivates their choice. Section 2.1 introduces the notation used to formulate a flow optimisation problem with constraints on the parameters. Subsequently section 2.2 describes several optimisation methods. The most suitable of these methods for turbulent flow optimisation in large parameter spaces are the gradient-based optimisation methods, to which section 2.3 is dedicated. The gradient-based methods require calculation of the gradient, which is challenging for turbulent flow optimisation in large parameter spaces, cfr. section 2.4. The adjoint method allows one to overcome this challenge (section 2.4.1). Finally section 2.5 introduces techniques to take constraints on the parameters into account.

2.1 Formulation of a flow optimisation problem

In this section, we introduce the notation used in this work to formulate a flow optimisation problem with constraints on the parameters.

Central to an optimisation problem is the formulation of a cost functional $\mathscr{J}$. The cost functional expresses the optimisation objective with a mathematical formula, the minimum of the functional is the optimal point. For flow optimisation, the cost functional typically measures a certain property of the flow $\mathscr{J}(\boldsymbol{q})$. Here, $\boldsymbol{q}(\mathbf{x}, t, \boldsymbol{\phi})$ is a turbulent flow solution ($\boldsymbol{q} = [\boldsymbol{u}, p]$, with $\boldsymbol{u}$ the velocity field, and p the pressure), which is defined on the spatial domain Ω in a time interval $[0, T]$ with $\mathbf{x} \in \Omega$. Here, T represents the time horizon, while t denotes the running time variable: $t \in [0, T]$. Turbulent flow optimisation is an instance of PDE-constrained optimisation, as to minimise $\mathscr{J}$ the flow field $\boldsymbol{q}$ is optimised in such a way that it still fulfils the Navier–Stokes equations and the Navier–Stokes boundary conditions, in other words $\boldsymbol{q}$ is constrained by partial differential equations (i.e. the Navier–Stokes equations). In general, part of the spatial and temporal Navier–Stokes boundary conditions, and part of the forcing are defined by the controls $\boldsymbol{\phi}$ (Bewley et al., 2001; Choi et al., 2008; Wei and Freund, 2006), they influence the solution of the Navier–Stokes problem $\boldsymbol{q}$ and are to be optimised. As a result, the cost functional depends on the parameters $\boldsymbol{\phi}$ through the flow solution $\boldsymbol{q}$: $\mathscr{J}(\boldsymbol{q}(\boldsymbol{\phi}))$.

Also the parameters $\boldsymbol{\phi}$ often have to respect certain limitations. If the limitations can be expressed mathematically independent of the solution of the Navier–Stokes problem, Hinze et al. (2009) categorises them as control constraints, in the other case they are categorised as state constraints. The state constraints make the optimisation problem computationally more demanding (see, e.g. Anderson and Venkatakrishnan, 1999; Giles and Pierce, 2000; Hinze et al., 2009). In the current work we focus on flow optimisation with control constraints, which is still challenging in large parameter space optimisation, as will be illustrated in the next chapters. The control constraints can be either an equality, or an inequality constraint. The latter imposes an upper or lower limit on a certain property of the controls. In this work only equality constraints are used. Flow optimisation problems with control constraints given by an equality are formulated as:

$$\min_{\boldsymbol{\phi}} \mathscr{J}(\boldsymbol{q}(\boldsymbol{\phi})) \quad \text{subject to} \quad \begin{cases} c_1(\boldsymbol{\phi}) = 0 \\ c_2(\boldsymbol{\phi}) = 0 \\ \vdots \\ c_n(\boldsymbol{\phi}) = 0 \end{cases}, \tag{2.1}$$

with $\boldsymbol{q}(\mathbf{x}, t, \boldsymbol{\phi})$ the flow solution corresponding to $\boldsymbol{\phi}$ governed by the Navier–Stokes equations for incompressible flows:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{2.2}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} + \frac{1}{\rho} \nabla p - \nu \nabla^2 \boldsymbol{u} - \boldsymbol{f} = 0, \tag{2.3}$$

with p the pressure field, ν the kinematic viscosity, and $\boldsymbol{f}$ a possible forcing term. Part of the spatial and temporal boundary conditions of this flow problem depend on the controls $\boldsymbol{\phi}$. As a result, the flow solution is a function of $\boldsymbol{\phi}$: $\boldsymbol{q}(\mathbf{x}, t, \boldsymbol{\phi})$. For simplicity of notation the flow solution is sometimes abbreviated as $\boldsymbol{q}(\boldsymbol{\phi})$ in the remaining part of the manuscript.

As this Navier–Stokes system has to be solved in order to evaluate the cost functional value for certain parameters $\boldsymbol{\phi}$, cost functional evaluations are costly with regard to calculation time. In order to limit the number of evaluations it is important to select the most suitable optimisation method, therefore several techniques used for turbulent flow optimisation are presented in section 2.2. First the flow problem is defined in function of dimensionless variables, which are used in the remainder of this book.

2.1.1 Formulation of a flow problem

The flow problem is formulated in function of the dimensionless variables, which are defined based on a set of reference values: L_{ref} the reference length, and u_{ref} the reference velocity. The corresponding dimensionless variables are (denoted with $+$):

$$x_i^+ = \frac{x_i}{L_{ref}} \tag{2.4}$$

$$t^+ = \frac{t\, u_{ref}}{L_{ref}} \tag{2.5}$$

$$u_i^+ = \frac{u_i}{u_{ref}} \tag{2.6}$$

$$p^+ = \frac{p}{\rho\, u_{ref}^2} \tag{2.7}$$

$$f^+ = \frac{f\, L_{ref}}{u_{ref}^2} \tag{2.8}$$

Also the time-space domain is defined with respect to the same reference values:

$$T^+ = \frac{T\, u_{ref}}{L_{ref}} \tag{2.9}$$

$$L_i^+ = \frac{L_i}{L_{ref}} \tag{2.10}$$

The dimensionless Navier-Stokes problem is $\mathscr{N}\, \boldsymbol{q}^+ = \boldsymbol{f}^+$:

$$\nabla \cdot \boldsymbol{u}^+ = 0, \tag{2.11}$$

$$\frac{\partial \boldsymbol{u}^+}{\partial t^+} + \boldsymbol{u}^+ \cdot \nabla \boldsymbol{u}^+ + \nabla p^+ - \frac{1}{Re}\nabla^2 \boldsymbol{u}^+ - \boldsymbol{f}^+ = 0, \tag{2.12}$$

with $Re = (u_{ref} \cdot L_{ref})/\nu$ the Reynolds number. For simplicity of notation, the superscript $+$ is omitted in the remainder of this text.

2.2 Flow optimisation methods

The selection of the optimisation technique depends on the properties of the optimisation problem. The optimisation parameters, for example, may be continuous, discrete, or a combination of continuous and discrete parameters (see, e.g. Nocedal and Wright, 2006). In the current work, optimisation is performed on continuous parameters and the optimisation is subject to the Navier–Stokes PDE constraint and control constraints. The techniques used for turbulent flow optimisation with or without control constraints belong in general to one of the following two groups: evolutionary algorithms (Kern, 2007; Jenn-Long, 2005; Fukagata et al., 2008; Epstein and Peigin, 2005; Hilgers and Boersma, 2001), and algorithms using the gradient and possibly the Hessian of the cost functional (Nocedal and Wright, 2006; Wei and Freund, 2006; Brezillon and Gauger, 2004; Hinze et al., 2009; Jameson and Vassberg, 2001). These methods are discussed next, neglecting the existence of control constraints. Section 2.5 comes back on the constraints.

The first group, the evolutionary algorithms, start from a population of parameters, evaluates their cost functional, called fitness function in this context, and subsequently generates a new population by performing operations on a selection of parameters from the previous population (Chong and Zak, 1996). The parameters from the previous population have a higher probability to be selected if they have a high fitness function value. There are two types of operations, called crossover and mutation (Chong and Zak, 1996). The crossover takes two parent parameters ϕ and exchanges parts of the parameters to generate new parameters. The new parameters are subsequently randomly changed with a certain probability during the operation called mutation. For more information, see e.g. Michalewicz (1996); Chong and Zak (1996).

Kern (2007) used several evolutionary algorithms and compared the number of functional evaluations before convergence of the optimisation for several test functionals. Parameter spaces with dimensions from 20 to 100 were tested, and optimisation in these spaces required a number of functional evaluations several factors higher than the number of dimensions. If this also applies to the current optimisation problem with a parameter space of dimension $\mathcal{O}(10^5)$ and large calculation cost for each evaluation of the cost functional, then the use of evolutionary algorithms is prohibited. This limits the suitable methods to the second group of optimisation techniques, the ones which use the gradient and possibly the Hessian of the cost functional.

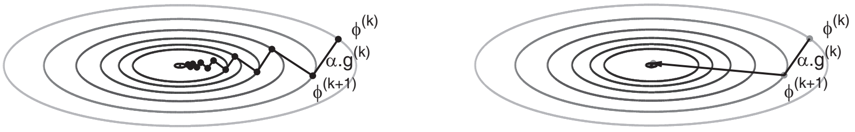
Nocedal and Wright (2006) divide the second group, using the gradient and possibly the Hessian, into line search algorithms and trust region methods. The first methods, line search algorithms, determine the search direction and then estimate the optimal step length in that direction. Subsequently a new search direction is determined and so on until a local minimum is found. For more information see, e.g. Luenberger (2003); Nocedal and Wright (2006); Chong and Zak (1996). The second method, the trust region method, first approximates the landscape of the cost functional with a model function based on the gradient and the Hessian or its approximation. A trust region radius is set and determines the maximal step. The new point is selected as the minimum of the model function within the trust region radius. For more details, see Luenberger (2003); Nocedal and Wright (2006); Conn, Gould, and Toint (2000).

This research focusses on optimisation of a large number of parameters for turbulent flows with up to $\mathcal{O}(10^5)$ parameters. This excludes optimisation techniques based on the Hessian or approximated Hessian, as calculation of the Hessian is too costly and storing it or its approximation is infeasible as it requires more than 150 Gb memory in case of $\mathcal{O}(10^5)$ parameters. Therefore trust region methods are excluded and we use line-search optimisation methods.

The line-search methods contain a group of methods that base their search direction on the gradient and another group of methods that also use the Hessian or its approximation. The latter are Newton (using the Hessian) and quasi-Newton methods (using the approximated Hessian). Although there exist quasi-Newton methods that circumvent storage of the approximated Hessian (for example the limited-memory BFGS method (see, e.g. Nocedal and Wright, 2006)), these techniques are not considered in the scope of this thesis. We opt for gradient-based line-search optimisation methods. These methods are further discussed in the next section.

2.3 Gradient-based line-search optimisation methods

Gradient-based line-search optimisation algorithms are iterative search methods, which start from an initial guess $\boldsymbol{\phi}^{(0)}$, and improve the cost functional by gradually descending through different steps $\boldsymbol{\phi}^{(k)}$ (Nocedal and Wright, 2006). As shown in figure 2.1, each iteration $\boldsymbol{\phi}^{(k)}$ is changed in the search direction $\boldsymbol{g}^{(k)}$ with a certain step length α to $\boldsymbol{\phi}^{(k+1)}$. The search direction $\boldsymbol{g}^{(k)}$ is based on the gradient $\mathcal{G}^{(k)}$ of the cost functional at the point $\boldsymbol{\phi}^{(k)}$. The step length α is determined with a line search such that $\boldsymbol{\phi}^{(k+1)} = \boldsymbol{\phi}^{(k)} + \alpha.\boldsymbol{g}^{(k)}$ leads to a minimum of $\mathscr{J}(\boldsymbol{q}(\boldsymbol{\phi}^{(k)} + \alpha.\boldsymbol{g}^{(k)}))$ as a function of α. In $\boldsymbol{\phi}^{(k+1)}$ a new search direction is determined and the cycle is repeated until a (local) optimum is reached (i.e., the gradient of the cost functional is zero). For simplicity of notation, the superscript (k) is omitted whenever possible in the remainder of this manuscript.



(a) (b)

Figure 2.1: Illustration of steepest descent method (a) and conjugate gradient method (b)

2.3.1 Search direction

The search direction $\boldsymbol{g}^{(k)}$ is gradient-based, requiring the gradient $\mathcal{G}^{(k)}$ of the cost functional at the point $\boldsymbol{\phi}^{(k)}$. There are several ways to choose $\boldsymbol{g}^{(k)}$, common directions are the steepest descent direction and the non-linear conjugate gradient directions, which are both explained in the next paragraphs.

The simplest gradient-based method is the steepest descent method, which uses as search direction the steepest descent direction $\boldsymbol{g}^{(k)} = -\mathcal{G}^{(k)}$ (see, e.g. Nocedal and Wright, 2006; Chong and Zak, 1996). This method can be extremely slow for difficult problems especially when the landscape is like an elongated valley (Nocedal and Wright, 2006; Chong and Zak, 1996, and see figure 2.1), because the search direction is always perpendicular to the previous search direction leading to an oscillatory behaviour in this type of landscapes.

The non-linear conjugate-gradient methods stem from linear conjugate-gradient methods, which are based on a Krylov-subspace method for the iterative solution of large linear systems of equations (Nocedal and Wright, 2006, p. 108). The non-linear conjugate gradient directions differ from the steepest descent direction in a term proportional to the previous search direction: $\boldsymbol{g}^{(k)} = -\mathcal{G}^{(k)} + \beta^{(k)}.\boldsymbol{g}^{(k-1)}$ (Nocedal and Wright, 2006; Chong and Zak, 1996). In general these methods are much more effective than the steepest descent method and almost as easy to compute (Nocedal and Wright, 2006). Bewley et al. (2001) and Wei and Freund (2006) performed flow optimisation with the Polak-Ribière non-linear conjugate gradient method. The Polak-Ribière method adapts the steepest descent direction such that the search direction is not every time perpendicular to the previous search direction, but if two subsequent gradients are almost equal, it changes the search direction towards the steepest descent direction (Press et al., 1996; Luenberger, 2003; Nocedal and Wright, 2006; Chong and Zak, 1996):

$$\mathbf{g}^{(k)} = -\mathcal{G}^{(k)} + \beta^{(k)}\mathbf{g}^{(k-1)}, \tag{2.13}$$

where the factor $\beta^{(k)}$ is given by $\beta^{(k)} = (\mathcal{G}^{(k)})^T(\mathcal{G}^{(k)} - \mathcal{G}^{(k-1)})/\|\mathcal{G}^{(k)}\|^2$. Figure 2.1 illustrates that the additional term prevents the optimisation from oscillatory behaviour like the steepest descent method in landscapes with an elongated valley.

The Polak-Ribière conjugate gradient method is selected here as it is considerably faster than the steepest-gradient descent method (Nocedal and Wright, 2006). The method requires the gradient of the cost functional, the calculation of which is not straightforward for large parameter spaces with expensive cost functional evaluations. The classical finite difference calculation of the gradient is infeasible since it requires one DNS simulation per parameter direction, as will be explained in detail in section 2.4. Therefore we resort to the more complex but more efficient adjoint method. Details on the adjoint method are given in Giles and Pierce (2000); Pironneau (1974); Jameson (1988); Jameson et al. (1998) and in section 2.4.1.

After the determination of $\boldsymbol{g}^{(k)}$ the step length α is required (Nocedal and Wright, 2006). To determine α typically a line search is used. The next section describes the standard *Brent* line search algorithm (Brent, 1973; Press et al., 1996) and the small changes to the algorithm in the current work.

2.3.2 Selection of step length: line-search algorithm

To find the optimal step length α, required for the Polak–Ribière method, a line-search is used. The step length α is optimal when it leads to a minimum for $\mathscr{J}(\boldsymbol{q}(\boldsymbol{\phi}^{(k)} + \alpha.\boldsymbol{g}^{(k)}))$. A robust, iterative, line search method, described in Press et al. (1996), is selected, which first brackets the minimum and subsequently isolates the minimum with the *Brent* algorithm (Brent, 1973; Press et al., 1996). The latter algorithm iteratively narrows down the interval in which it is known that the minimum is located in such a way that the smaller interval still contains the minimum.

The determination of the bracketing interval will be discussed first, followed by the *Brent* algorithm.

Search bracketing interval

The algorithm used to determine the interval is based on the *mnbrak* algorithm described in Press et al. (1996). It has been adapted to take the constraints into account (see section 2.5) and to reuse as much information about the cost functional as possible to reduce the number of cost functional evaluations. The adaptations are discussed after presentation of the standard algorithm found in Press et al. (1996).

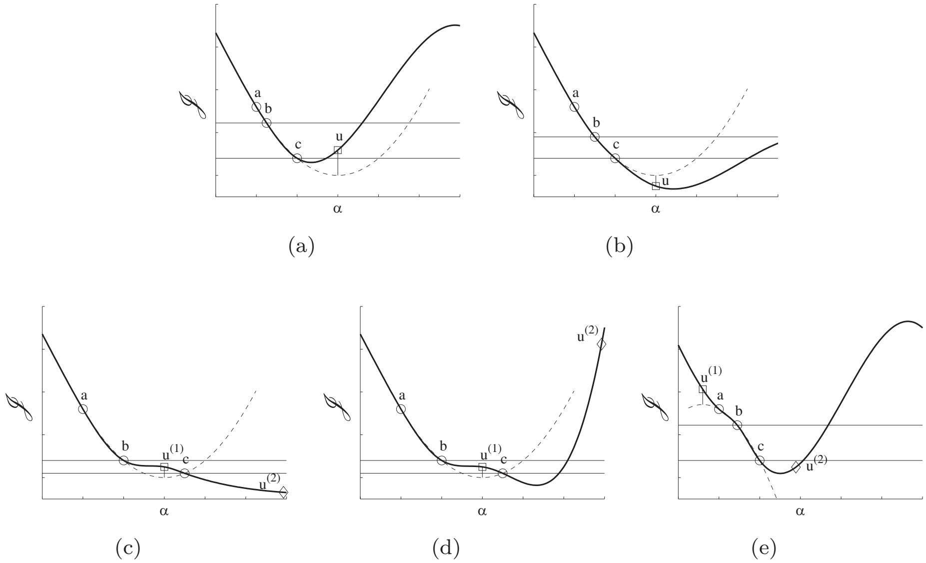
The algorithm searches an interval that brackets the minimum. A bracketing interval is defined when three step lengths a, b, and c are found with $b \in]a, c[$, $\mathscr{J}_b < \mathscr{J}_a$, and $\mathscr{J}_b < \mathscr{J}_c$. The *mnbrak* algorithm (Press et al., 1996) iteratively adapts the values of a, b and c. As long as the interval does not include a minimum, a step length u is selected that is likely to be able to replace a, b or c such that the new interval brackets a minimum or is shifted in the downhill direction. In the latter case, the process is repeated starting from the new interval. The test step length u is first determined by parabolic extrapolation: $u^{(1)}$, see figure 2.2 (a-e) and if this fails (as is the case in figures 2.2 (c,d,e)), a default magnification is used to select u: $u^{(2)} = c + 1.618(c - b)$ with 1.618 the golden section number $((1 + \sqrt{5})/2)$.

The parabolic extrapolation assumes a parabolic function behaviour and selects u to be the minimum of the parabola approximating the functional. The parabolic extrapolation step is useful if it leads to bracketing a minimum (for example in fig. 2.2 (a) with $b < c < u$, $\mathscr{J}_c < \mathscr{J}_b$ and $\mathscr{J}_c < \mathscr{J}_u$) or if it shifts the interval in the downhill direction (for example in fig. 2.2 (b) the new interval is $[b, u]$). The parabolic extrapolation is not used in three situations:

1. if the extrapolation is larger than an upper limit $u > b + 100(c - b)$.
2. if, like in figure 2.2 (c,d), u is in the interval $]b, c[$ but does not lead to a bracketing interval as $\mathscr{J}_u \in]\mathscr{J}_c, \mathscr{J}_b[$ (if $\mathscr{J}_u$ is outside this interval, the minimum is bracketed)
3. if the extremum of the parabola fitted through a, b, c is on the uphill side of b because the parabola has a negative second derivative as illustrated in figure 2.2 (e).

In case (1) the maximum extrapolation is used to define $u = b + 100(c - b)$, in case (2) the algorithm switches to the default magnification (fig. 2.2 (c,d)), and in case (3) $\mathscr{J}_{u^{(1)}}$ is not evaluated, instead the default magnification is used to define $u = u^{(2)}$ (fig. 2.2 (e)). The default magnification step increases the interval in the downhill direction, as is the case in Figure 2.2 (c,e), or leads to an interval containing a minimum, see figure 2.2 (d).

The *mnbrak* algorithm described in Press et al. (1996) uses 3 step lengths for the parabolic approximation, as a consequence two step lengths b and c are selected and the corresponding cost functionals are evaluated before the first iteration of the algorithm. In case of gradient based optimisation, it is also possible to base the parabolic approximation on two step lengths and the already available gradient, eliminating one cost functional evaluation necessary to prepare the first iteration. As the cost functional evaluations are computationally expensive, the algorithm has been extended in this way. The gradient information is used for the parabolic approximation as long as the step length a, b or c is equal to zero.



(a) (b)

(c) (d) (e)

Figure 2.2: *mnbrak* algorithm applied to generic function $\mathscr{J}(\alpha)$ (—). New step length u calculated with parabolic extrapolation (□). Parabolic approximation (−−) based on a, b, and c (○) (a,b) parabolic approximation is successful. (c,d,e) parabolic extrapolation $u^{(1)}$ fails because (c,d) $u^{(1)} \in]b, c[$ and $\mathscr{J}_{u^{(1)}} \in]\mathscr{J}_c, \mathscr{J}_b[$, (e) the extremum of the parabola fitted through a, b, c is on the uphill side of b . (c,d,e) The default magnification is used to determine step $u^{(2)}$ (◊). $u^{(2)}$ enlarges the interval (c,e), or $u^{(2)}$ results in interval (d).

Search step length: ***Brent*** algorithm

The interval selected by the *mnbrak* algorithm is subsequently decreased in size by the *Brent* algorithm described in Press et al. (1996). This algorithm decreases the interval $[a, b, c]$ iteratively until the minimum is located with the required tolerance *tol* and this in such a way that the interval still contains a minimum. It therefore combines two line search techniques: inverse parabolic interpolation (see, e.g. Press et al., 1996; Brent, 1973) and the golden section method (see, e.g. Press et al., 1996; Chong and Zak, 1996).

Inverse parabolic interpolation fits a parabola through three points, evaluates the cost functional at the parabola's minimum and if possible narrows down the interval. As points to fit the parabola through, the Brent algorithm selects the three evaluated step lengths that lead to the smallest cost functional value. These step lengths are denoted as x, v, and w. This method is fast if the function

$\mathscr{J}(\boldsymbol{q}(\boldsymbol{\phi}^{(k)} + \alpha.\boldsymbol{g}^{(k)}))$ is a parabolic function of α, but it is not fail-safe for all function behaviours (see Press et al., 1996). The Brent algorithm therefore subjects the step length $u^{(1)}$ proposed by the inverse parabolic interpolation to some tests. If the step length fails one of these tests, the step length is considered unacceptable and the fail safe, but slower, golden section method is used instead. First the tests are described, afterwards the procedure for successful step lengths is briefly introduced.

The Brent algorithm (Press et al., 1996) considers the step length $u^{(1)}$ proposed by the inverse parabolic interpolation unacceptable in three situations:

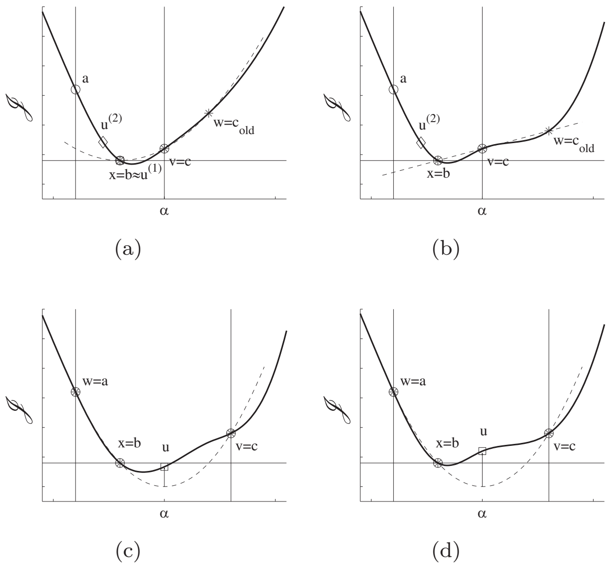
1. if, like in figure 2.3 (a), the distance between u and, a, b, or c is not large enough, such that calculating $\mathscr{J}_u$ would not give new information,
2. if u is outside the interval like in figure 2.3 (b) where $u^{(1)}$ is far to the left,
3. if the inverse parabolic interpolation does not seem to converge. To this end the distance between u and b is monitored. If u's distance to b does not converge to zero (current $|u, b|$ bigger than the distance $|u, b|$ in the second but last iteration) the step length u is unacceptable.

In these situations the cost functional value for the proposed $u^{(1)}$ is not calculated, and the Brent algorithm switches to the very stable golden section method.

If, on the other hand, the step length u proposed by the inverse parabolic interpolation is considered acceptable, $\mathscr{J}_u$ is calculated and it is decided how to proceed. Based on the functional values of a, b, c and u the smallest interval $[\alpha_1, \alpha_2, \alpha_3]$ is selected for which $\mathscr{J}_{\alpha_2} < min(\mathscr{J}_{\alpha_1}, \mathscr{J}_{\alpha_3})$. There are two types of situations depending on the value of $\mathscr{J}_u$ compared to $\mathscr{J}_b$, illustrated in figures 2.3(c,d). Figure 2.3 (c) is an example of $\mathscr{J}_u < \mathscr{J}_b$, the interval subsequently is narrowed to $[b, u, c]$. (Note that if u had been at the other side of b, the new interval would be $[a, u, b]$.) Figure 2.3 (d) depicts a situation where $\mathscr{J}_u > \mathscr{J}_b$. In this situation the interval is narrowed to $[a, b, u]$. (Note that if u had been at the other side of b, the new interval would be $[u, b, c]$).

The golden section method (Brent, 1973; Press et al., 1996; Chong and Zak, 1996) evaluates the cost functional for $u = b + 0.382 \max(a - b, c - b)$. Based on the functional values of a, b, c and u the smallest interval $[\alpha_1, \alpha_2, \alpha_3]$ is selected for which $\mathscr{J}_{\alpha_2} < min(\mathscr{J}_{\alpha_1}, \mathscr{J}_{\alpha_3})$.

The *Brent* algorithm tries during each iteration first to narrow down the interval with inverse parabolic interpolation. If the parabolic fit gives an unacceptable value for u, the golden section method is used. This is repeated until the interval is small enough. The stop criteria imposes a relative accuracy smaller than the tolerance *tol* (set to 2×10^{-6} in this work). The relative accuracy is calculated as follows: b is assumed to be the step length that leads to the minimum, the maximum error



(a) (b)

(c) (d)

Figure 2.3: *Brent* algorithm applied to generic function $\mathscr{J}(\alpha)$ (—). New step length u calculated with inverse parabolic interpolation ($\Box$). Parabolic approximation ($--$) based on x, v, and w ($*$). Step lengths determining interval a, b, c ($\circ$) (a,b) inverse parabolic approximation is rejected, step length is calculated with golden section method ($\Diamond$), because minimum parabolic fit is (a) too close to b (b) outside interval. (c,d) inverse parabolic approximation is successful. (c) $\mathscr{J}_u < \mathscr{J}_b$, the interval is narrowed to $[b, u, c]$. (d) $\mathscr{J}_u > \mathscr{J}_b$, interval is narrowed to $[a, b, u]$

on the minimum is then $\max(|b-a|, |c-b|)$. The relative accuracy is this error divided by $|b|$.

In the *Brent* algorithm described in Press et al. (1996) the parabolic interpolation is based on the three step lengths leading to the three most optimal values for the cost functional found so far during the line search.

In the current work the *Brent* algorithm is extended to use the gradient information for the parabolic interpolation as long as it is still available in the interval in analogy with the modification of the *mnbrak* algorithm.

2.4 Gradient of the cost functional

As discussed in the introduction, it is challenging to calculate the gradient of the cost functional for large parameter spaces and cost functional evaluations requiring expensive PDE simulations. This is clarified here for a generic cost functional. The remedy provided by the adjoint method is discussed. In the literature two approaches, continuous and discrete adjoint, are discussed (see, for example, Giles and Pierce, 2000; Bewley et al., 2001; Jameson and Vassberg, 2001). An overview of the advantages and disadvantages of both approaches is given. Section 2.4.1 elaborates on the continuous adjoint method used here to calculate the gradient. The adjoint boundary conditions are also discussed. Part of this discussion is also published in Delport, Baelmans, and Meyers (2009)

First of all we define a generic cost functional $\mathscr{J}$, for which the calculation of the gradient is discussed:

$$\begin{aligned}\mathscr{J}(\boldsymbol{q}) &= \mathscr{J}_{\Omega}(\boldsymbol{q}) + \mathscr{J}_{\partial\Omega}(\boldsymbol{q}) + \mathscr{J}_{T}(\boldsymbol{q}) \qquad (2.14)\\ &= \int_0^T \int_{\Omega} J_{\Omega}(\boldsymbol{q})\,\mathrm{d}\mathbf{x}\,\mathrm{d}t + \int_0^T \int_{\partial\Omega} J_{\partial\Omega}(\boldsymbol{q})\,\mathrm{d}\mathbf{x}\,\mathrm{d}t + \int_{\Omega} J_T(\boldsymbol{q})\,\mathrm{d}\mathbf{x},\end{aligned}$$

with $\mathscr{J}_{\Omega}$, $\mathscr{J}_{\partial\Omega}$, and $\mathscr{J}_{T}$ contributions to the cost functional respectively depending on $\boldsymbol{q}$ in the full space-time domain $\Omega \times [0, T]$, $\boldsymbol{q}$ on the spatial boundaries $\partial\Omega$ of that domain, and $\boldsymbol{q}$ on the temporal boundary at time T. The symbols J denote the underlying operators of $\mathscr{J}$ and stand for functionals depending on (part of) the flow field $\boldsymbol{q}$.

For the derivation of the gradient, it is useful to introduce the usual L^2 inner product between two tensors $\boldsymbol{a}, \boldsymbol{b} \in L^2(\Omega \times [0, T])$, as

$$(\boldsymbol{a}, \boldsymbol{b}) \equiv \int_0^T \int_{\Omega} \boldsymbol{a}(\mathbf{x}, t) \cdot \boldsymbol{b}(\mathbf{x}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}t, \qquad (2.15)$$

where $\boldsymbol{a} \cdot \boldsymbol{b} = a_i b_i$ with $a_i(\mathbf{x}, t), b_i(\mathbf{x}, t)$ the components of $\boldsymbol{a}, \boldsymbol{b}$, and using Einstein's summation convention.

In order to compute the gradient of the cost functional, we need to evaluate the sensitivity of $\mathscr{J}$ at a given point $\boldsymbol{q}(\boldsymbol{\phi}) = \boldsymbol{q}(\mathbf{x}, t, \boldsymbol{\phi})$ to small changes $\delta\boldsymbol{\phi}$ on the control parameters $\boldsymbol{\phi}$ (with, cfr. section 2.1, $\boldsymbol{\phi}$ defining part of the spatial and temporal boundary conditions and part of the forcing). The sensitivity is calculated as the Gateaux-differential of $\mathscr{J}$ in the point $\boldsymbol{q}(\boldsymbol{\phi})$ (Luenberger, 1969). For the first term $\mathscr{J}_{\Omega}$ of the cost functional in equation (2.14), the Gateaux differential is:

$$
\begin{aligned}
\delta\mathscr{J}_\Omega(\boldsymbol{\phi};\delta\boldsymbol{\phi}) &\equiv \frac{\mathrm{d}}{\mathrm{d}\alpha}\mathscr{J}_\Omega\big(\boldsymbol{q}(\boldsymbol{\phi}+\alpha\delta\boldsymbol{\phi})\big)\Big|_{\alpha=0} && (2.16)\\
&= \int_0^T\int_\Omega \frac{\partial J_\Omega(\boldsymbol{q}(\boldsymbol{\phi}))}{\partial\boldsymbol{q}}.\ \frac{\partial\boldsymbol{q}(\boldsymbol{\phi}+\alpha.\delta\boldsymbol{\phi})}{\partial\alpha}\Big|_{\alpha=0}\ \mathrm{d}\mathbf{x}\ \mathrm{d}t && (2.17)\\
&= \left(\frac{\partial J_\Omega(\boldsymbol{q}(\boldsymbol{\phi}))}{\partial\boldsymbol{q}},\delta\boldsymbol{q}\right), && (2.18)
\end{aligned}
$$

with $\delta\boldsymbol{q} = [\delta\boldsymbol{u},\delta p]$ the sensitivity of the flow solution to a change of $\boldsymbol{\phi}$ in the direction $\delta\boldsymbol{\phi}$. The Gateaux differentials of the other two components of $\mathscr{J}$ (2.14) are defined similar to equation (2.16):

$$
\begin{aligned}
\delta\mathscr{J}_{\partial\Omega}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) &\equiv \frac{\mathrm{d}}{\mathrm{d}\alpha}\left(\int_0^T\int_{\partial\Omega} J_{\partial\Omega}(\boldsymbol{q}(\boldsymbol{\phi}+\alpha\delta\boldsymbol{\phi}))\ \mathrm{d}\mathbf{x}\ \mathrm{d}t\right)\Bigg|_{\alpha=0} && (2.19)\\
\delta\mathscr{J}_{T}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) &\equiv \frac{\mathrm{d}}{\mathrm{d}\alpha}\left(\int_0^T\int_{\partial\Omega} J_{T}(\boldsymbol{q}(\boldsymbol{\phi}+\alpha\delta\boldsymbol{\phi}))\ \mathrm{d}\mathbf{x}\ \mathrm{d}t\right)\Bigg|_{\alpha=0} && (2.20)
\end{aligned}
$$

The central issue in the calculation of equation (2.18) is the determination of $\delta\boldsymbol{q}$. $\delta\boldsymbol{q}$ is governed by the Navier–Stokes equations (2.11–2.12). For small changes $\delta\boldsymbol{q}$, the Navier–Stokes equations can be linearised around the solution $\boldsymbol{q}(\boldsymbol{\phi})$. The linear partial differential equations are noted as:

$$\mathscr{N}'(\boldsymbol{q})\ \delta\boldsymbol{q} = \boldsymbol{f}', \tag{2.21}$$

where $\mathscr{N}'(\boldsymbol{q})$ is a linear PDE operator which depends on $\boldsymbol{q}(\boldsymbol{\phi})$ and acts on $\delta\boldsymbol{q}$. As the boundary conditions of this linearised problem depend on $\delta\boldsymbol{\phi}$, this system has to be solved for each parameter direction. When $\boldsymbol{\phi}$ contains a large number of degrees of freedom, the determination of $\delta\boldsymbol{q}$, for each $\delta\boldsymbol{\phi}$, becomes prohibitive. This problem can be circumvented by formulating an adjoint problem. Solving the adjoint problem is about as costly as a flow simulation. The gradient can then be determined at a negligible cost based on the solution of only one adjoint problem (see, e.g., Giles and Pierce, 2000; Bewley et al., 2001; Jameson and Vassberg, 2001, and in the following section 2.4.1).

In the literature, two methodologies exist for adjoint-based gradient estimation, i.e. the continuous approach, and the discrete approach (Giles and Pierce, 2000; Anderson and Venkatakrishnan, 1999). In the discrete approach, first the PDE system (here the incompressible Navier–Stokes equations) is discretised, subsequently this system is linearised, and finally the discrete adjoint of this linear system is formulated (see path at the bottom of fig. 2.4). This results in the adjoint code, here denoted as D. In this thesis, the continuous approach is used. This method first formulates the adjoint equations before discretising them (see path at the top of fig. 2.4). The obtained code is depicted as code C in figure 2.4. In the limit of vanishing grid spacing, code C, and D should lead to the same gradient (Giles and Pierce, 2000; Burkardt, Gunzburger, and Peterson, 2002; Anderson and Venkatakrishnan, 1999), for finite grids, however, there is a difference. The discrete approach calculates the gradient of the discrete cost functional while the continuous approach determines the gradient of the continuous system analytically and subsequently approximates it discretely. The latter is not fully consistent with the gradient of the discrete cost functional such that in the optimum of the discrete cost functional the continuous adjoint-based gradient may be not exactly zero or orthogonal to the constraint surface (Giles and Pierce, 2000; Burkardt et al., 2002). The difference stems from a difference in the discretisation used in the two codes C and D. The discretisation in code D is entirely determined by the discretisation of the PDE system. This has the advantage that, the adjoint code can be created automatically by automatic differentiation (AD) of the forward code (i.e. DNS code). AD tools have been applied to generate adjoint codes of CFD codes (see, e.g. Giering, Kaminski, and Slawig, 2005; Burkardt et al., 2002), but it is not yet guaranteed that no manual interference is necessary (Giering et al., 2005; Hay, Pelletier, and Di Caro, 2009). The continuous approach, leading to code C, allows more flexibility in the selection of discretisation schemes for the different equations, additionally it is known to require less memory (Giles and Pierce, 2000), and in our case it helps to significantly limit the amount of disk storage. The last issue is further discussed in section 3.2.2. A further advantage of the continuous method is that it gives more physical insight into adjoint variables and the role of the adjoint boundary conditions is much clearer (Giles and Pierce, 2000). The continuous adjoint method is further detailed in the next section.

2.4.1 Continuous adjoint method

The current subsection reviews the continuous adjoint-based method for the calculation of gradients (Pironneau, 1974; Jameson, 1988; Jameson et al., 1998)

The adjoint method is based on the adjoint identity (cf, e.g., Bewley et al., 2001; Giles and Pierce, 1997). The identity defines the adjoint (linear) PDE operator $\mathscr{N}^*(\boldsymbol{q})$ by requiring that:

$$(\boldsymbol{q}^*, \mathscr{N}'\ \delta\boldsymbol{q}) \equiv (\mathscr{N}^*\boldsymbol{q}^*, \delta\boldsymbol{q}) + BT, \tag{2.22}$$

where BT are boundary terms which arise when $\mathscr{N}'$ on the left hand side of (2.22) is changed into $\mathscr{N}^*$ on the right-hand side of (2.22) by means of integration by

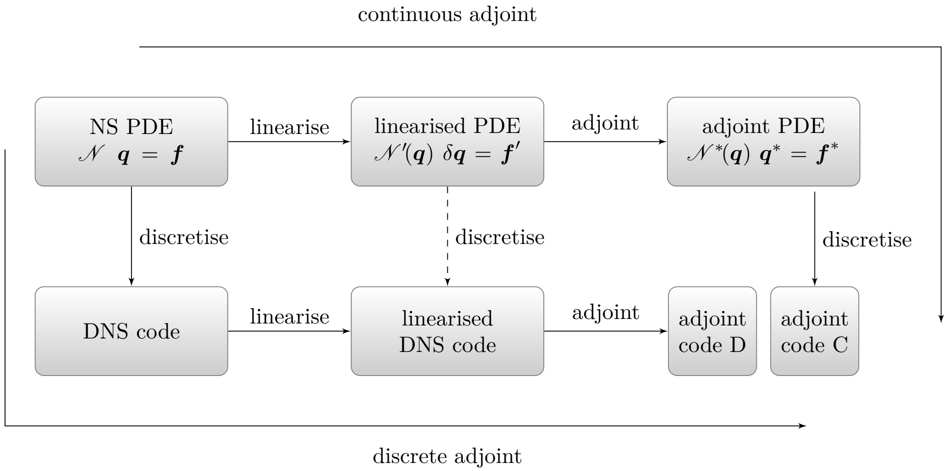


Figure 2.4: Continuous and discrete approach to derive the adjoint code, graph based on figure in Giles and Pierce (2000)

parts. The adjoint operator $\mathscr{N}^*$ is used to formulate a set of linear PDE's, i.e.

$$\mathscr{N}^*(\boldsymbol{q})\ \boldsymbol{q}^* \quad = \quad \boldsymbol{f}^*. \tag{2.23}$$

By selecting the source term $\boldsymbol{f}^*$ in (2.23) equal to $\partial J_\Omega(\boldsymbol{q})/\partial\boldsymbol{q}$, the solution $\boldsymbol{q}^* = [u_1^*, u_2^*, u_3^*, p^*]$, to this set of PDE's can be used to express the Gateaux-differential in (2.18), since in that case (using (2.21-2.23)):

$$\delta\mathscr{J}_\Omega(\boldsymbol{\phi};\delta\boldsymbol{\phi}) = \left(\frac{\partial J_\Omega(\boldsymbol{q})}{\partial\boldsymbol{q}}, \delta\boldsymbol{q}\right) = (\boldsymbol{f}^*, \delta\boldsymbol{q}) = \big(\boldsymbol{q}^*, \boldsymbol{f}'\big) - BT. \tag{2.24}$$

The Gateaux-differential of the total cost functional is then:

$$\delta\mathscr{J}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) = (\boldsymbol{q}^*, f') - BT + \delta\mathscr{J}_{\partial\Omega}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) + \delta\mathscr{J}_T(\boldsymbol{\phi};\delta\boldsymbol{\phi}). \tag{2.25}$$

The essential advantage of the adjoint formulation is that a careful selection of boundary conditions of (2.23) makes the Gateaux differential (2.25) directly dependent of $\delta\boldsymbol{\phi}$ instead of implicitly through $\delta\boldsymbol{q}(\mathbf{x}, t, \boldsymbol{\phi}, \delta\boldsymbol{\phi})$. The explicit dependence of $\delta\mathscr{J}$ on $\delta\boldsymbol{\phi}$ appears as internal products of $\boldsymbol{q}^*$ with $\delta\boldsymbol{\phi}$-dependent terms (such as, e.g., $\boldsymbol{f}'$ in eq. (2.24)). Since the adjoint problem is independent of the direction $\delta\boldsymbol{\phi}$, and the cost for the calculation of the adjoint problem is similar to the cost for the calculation of the flow problem itself, this method is much more efficient than directly calculating $\delta \boldsymbol{q}$ through (2.21) for every direction $\delta \boldsymbol{\phi}$.

The boundary conditions for the adjoint PDE's (2.23) are identified by requiring that $-BT + \delta\mathscr{J}_{\partial\Omega} + \delta\mathscr{J}_T$ (on the right-hand side of eq. (2.25)) is independent of $\delta \boldsymbol{q}$ that are unknown without a linearised flow simulation. The boundary conditions are determined in two steps. Firstly the adjoint equations and the boundary terms are derived based on eq. (2.22), secondly the adjoint boundary conditions are determined. The principle is illustrated for the incompressible Navier–Stokes equations with the generic cost functional (2.14).

Derivation of adjoint equations of the Navier–Stokes equations

The adjoint state equations $\mathscr{N}^*(\boldsymbol{q})q^* = f^*$ are defined such that equation (2.22) is fulfilled. The linearised Navier–Stokes equations $\mathscr{N}'(\boldsymbol{q})\delta\boldsymbol{q} = f'$ are:

$$\begin{aligned} \frac{\partial u_i'}{\partial t} + \frac{\partial u_i' u_j}{\partial x_j} + \frac{\partial u_i u_j'}{\partial x_j} + \frac{\partial p'}{\partial x_i} - \frac{1}{Re}\left(\frac{\partial^2 u_i'}{\partial x_j x_j} + \frac{\partial^2 u_j'}{\partial x_i x_j}\right) &= f_i' \qquad i = 1,2,3 \\ \frac{\partial u_j'}{\partial x_j} &= 0 \end{aligned} \tag{2.26}$$

with summation over the repeated indices j, $\boldsymbol{q} = [u_1, u_2, u_3, p]$, and $\delta\boldsymbol{q} = [u_1', u_2', u_3', p']$.

The adjoint equations $\mathscr{N}^*(\boldsymbol{q})\boldsymbol{q}^* = f^*$ can be derived, using integration by parts, based on the adjoint identity (2.22) and the linearised Navier-Stokes equations

(2.26):

$$
\begin{aligned}
&(\mathscr{N}'(\boldsymbol{q})\delta\boldsymbol{q},\boldsymbol{q}^*) \qquad (2.27)\\
= \quad & \int_0^T \int_\Omega \left(\frac{\partial u_i'}{\partial t} + \frac{\partial (u_i' u_j + u_i u_j')}{\partial x_j} + \frac{\partial p'}{\partial x_i} - \frac{\partial}{\partial x_j}\frac{1}{Re}\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right)\right) u_i^* \,\mathrm{d}\mathbf{x}\mathrm{d}t\\
+ \quad & \int_0^T \int_\Omega \frac{\partial u_j'}{\partial x_j} p^* \,\mathrm{d}\mathbf{x}\mathrm{d}t\\
= \quad & \int_\Omega u_i' u_i^* |_0^T \,\mathrm{d}\mathbf{x} + \int_0^T \int_\Omega -u_i' \frac{\partial u_i^*}{\partial t} \,\mathrm{d}\mathbf{x}\mathrm{d}t\\
+ \quad & \int_0^T \int_{\partial\Omega} \left((u_i' u_j + u_i u_j') u_i^* n_j + p' u_i^* n_i - \frac{1}{Re}\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) u_i^* n_j \right) \mathrm{d}\mathbf{x}\mathrm{d}t\\
+ \quad & \int_0^T \int_\Omega -(u_i' u_j + u_i u_j') \frac{\partial u_i^*}{\partial x_j} - p' \frac{\partial u_i^*}{\partial x_i} + \frac{1}{Re}\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) \frac{\partial u_i^*}{\partial x_j} \,\mathrm{d}\mathbf{x}\mathrm{d}t\\
+ \quad & \int_0^T \int_{\partial\Omega} u_j' p^* n_j \mathrm{d}\mathbf{x}\mathrm{d}t - \int_0^T \int_\Omega u_j' \frac{\partial p^*}{\partial x_j} \,\mathrm{d}\mathbf{x}\mathrm{d}t
\end{aligned}
$$

With the $\partial\Omega$ integrals integrating over the spatial boundary $\partial\Omega$, $\mathbf{n} = [n_1, n_2, n_3]$ the normal on the boundary $\partial\Omega$, and summation over repeated indices i, j.

A second integration by parts removes the derivatives on u_i', and u_j' in the second $(1/Re)$-term:

$$
\begin{aligned}
&(\mathscr{N}'(\boldsymbol{q})\delta\boldsymbol{q},\boldsymbol{q}^*) \qquad (2.28)\\
= \quad & \int_\Omega u_i' u_i^* |_0^T \,\mathrm{d}\mathbf{x} - \int_0^T \int_\Omega u_i' \frac{\partial u_i^*}{\partial t} \,\mathrm{d}\mathbf{x}\mathrm{d}t\\
+ \quad & \int_0^T \int_{\partial\Omega} \left((u_i' u_j + u_i u_j') u_i^* + p' u_j^* - \frac{1}{Re}\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) u_i^* \right) n_j \mathrm{d}\mathbf{x}\mathrm{d}t\\
+ \quad & \int_0^T \int_\Omega -(u_i' u_j + u_i u_j') \frac{\partial u_i^*}{\partial x_j} - p' \frac{\partial u_i^*}{\partial x_i} - \frac{1}{Re}\left(u_i' \frac{\partial^2 u_i^*}{\partial x_j \partial x_j} + u_j' \frac{\partial^2 u_i^*}{\partial x_i \partial x_j} \right) \,\mathrm{d}\mathbf{x}\mathrm{d}t\\
+ \quad & \int_0^T \int_{\partial\Omega} \left(\frac{1}{Re} u_i' \left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right) n_j \right) \mathrm{d}\mathbf{x}\mathrm{d}t\\
+ \quad & \int_0^T \int_{\partial\Omega} u_j' p^* n_j \mathrm{d}\mathbf{x}\mathrm{d}t - \int_0^T \int_\Omega u_j' \frac{\partial p^*}{\partial x_j} \,\mathrm{d}\mathbf{x}\mathrm{d}t
\end{aligned}
$$

Corresponding with equation (2.22), the integral over the entire space and time domain is reformulated in the shape of $(\delta\boldsymbol{q}, \mathscr{N}^*(\boldsymbol{q})\boldsymbol{q}^*)$ and the terms that are only evaluated at the temporal or spatial boundaries of the domain are grouped as the boundary terms BT.

$$
\begin{aligned}
&(\mathscr{N}'(\boldsymbol{q})\delta\boldsymbol{q}, \boldsymbol{q}^*) && (2.29)\\
&= \int_0^T \int_\Omega u_i' \left(-\frac{\partial u_i^*}{\partial t} - u_j \frac{\partial u_i^*}{\partial x_j} - u_j \frac{\partial u_j^*}{\partial x_i} \right) \mathrm{d}\mathbf{x}\mathrm{d}t \\
&\quad + \int_0^T \int_\Omega u_i' \left(-\frac{\partial p^*}{\partial x_i} - \frac{1}{Re} \left(\frac{\partial^2 u_i^*}{\partial x_j \partial x_j} + \frac{\partial^2 u_j^*}{\partial x_j \partial x_i} \right) \right) \mathrm{d}\mathbf{x}\mathrm{d}t \\
&\quad + \int_0^T \int_\Omega -p' \frac{\partial u_i^*}{\partial x_i} \, \mathrm{d}\mathbf{x}\mathrm{d}t + BT \\
&= (\delta\boldsymbol{q}, \mathscr{N}^*(\boldsymbol{q})\boldsymbol{q}^*) + BT && (2.30)
\end{aligned}
$$

Following the definition of the adjoint identity (eq. (2.22)), $\mathscr{N}^*(\boldsymbol{q})q^*$ is (see also, e.g., Bewley et al., 2001):

$$
-\frac{\partial u_i^*}{\partial t} - u_j \frac{\partial u_i^*}{\partial x_j} - u_j \frac{\partial u_j^*}{\partial x_i} - \frac{1}{Re} \left(\frac{\partial^2 u_i^*}{\partial x_j \partial x_j} + \frac{\partial^2 u_j^*}{\partial x_j \partial x_i} \right) - \frac{\partial p^*}{\partial x_i} = f_i^* \quad (2.31)
$$

$$
-\frac{\partial u_i^*}{\partial x_i} = f_4^* \quad (2.32)
$$

with $\boldsymbol{f}^* = \partial \mathscr{J}_\Omega / \partial \boldsymbol{q}$, cfr. equation (2.24).

Two further remarks about the adjoint equations are important. Firstly the sign of the time derivative, and the convective terms in the adjoint equations are opposite to the forward equations (2.11-2.12). Hence, the characteristics in the adjoint equations propagate in reversed direction. As a result, the adjoint problem needs to be calculated back in time (Templeton, Wang, and Moin, 2006, and see figure 2.5). Secondly, during the backward calculation the solution of the forward (Navier–Stokes) simulation ($\boldsymbol{u}$) is required. The solution has to be stored onto disk.

Derivation of boundary conditions for the adjoint Navier–Stokes equations

The adjoint boundary conditions are derived such that the Gateaux differential of the cost functional using the adjoint formulation is formulated through $\delta\boldsymbol{\phi}$, $\boldsymbol{q}(\boldsymbol{\phi})$,

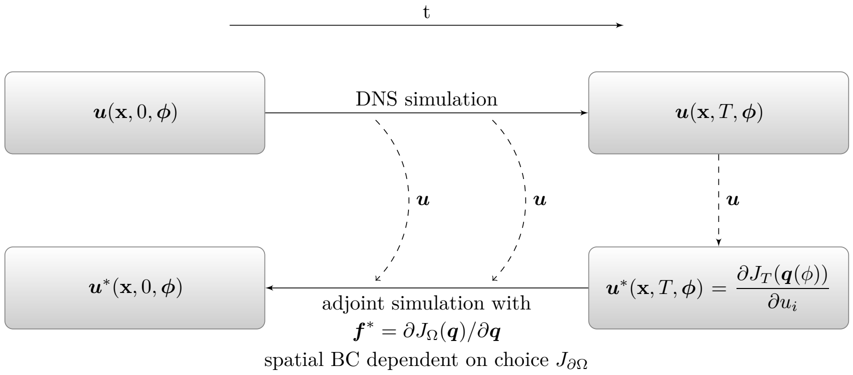


Figure 2.5: Impact of the choice of the cost functional on the adjoint boundary conditions. Input of $\boldsymbol{u}$ in the adjoint simulation

and $\boldsymbol{q}^*$ and does not require the calculation of $\delta\boldsymbol{q}$. The Gateaux differential (2.25) can be divided into parts that integrate over the same boundary, i.e. $t = 0$, $t = T$, or $\partial\Omega$. The requirement that each of these parts should not require the calculation of $\delta\boldsymbol{q}$ leads to the boundary conditions for the adjoint equations, as will be detailed in this section.

For sake of a clear notation, we define some additional L^2 inner products integrating over parts of the domain, in analogy with (2.15):

$$(\boldsymbol{a}, \boldsymbol{b})_{\partial\Omega} = \int_0^T \int_{\partial\Omega} \boldsymbol{a}(\mathbf{x}, t).\boldsymbol{b}(\mathbf{x}, t)\ \mathrm{d}\mathbf{x}\ \mathrm{d}t \tag{2.33}$$

$$(\boldsymbol{a}, \boldsymbol{b})_{t_i} = \int_\Omega \boldsymbol{a}(\mathbf{x}, t_i).\boldsymbol{b}(\mathbf{x}, t_i)\ \mathrm{d}\mathbf{x} \tag{2.34}$$

The boundary terms BT in (2.25) originate from the use of partial integration (2.27-2.28). These terms can be split in three parts according to the part of the boundary over which they integrate, i.e.

$$BT_0 = -\left(\boldsymbol{u}^*, \delta\boldsymbol{u}\right)_0 \tag{2.35}$$

$$BT_T = \left(\boldsymbol{u}^*, \delta\boldsymbol{u}\right)_T \tag{2.36}$$

$$\begin{aligned} BT_{\partial\Omega} &= \int_0^T \int_{\partial\Omega} \left((u_i' u_j + u_i u_j')u_i^* + p' u_j^* + u_j' p^*\right) n_j \mathrm{d}\mathbf{x}\mathrm{d}t \\ &+ \int_0^T \int_{\partial\Omega} \frac{1}{Re} \left(-\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) u_i^* + u_i' \left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i}\right)\right) n_j \mathrm{d}\mathbf{x}\mathrm{d}t \end{aligned} \tag{2.37}$$

The Gateaux differential (2.25) with the terms gathered according to their integration domain is:

$$\begin{aligned} \delta\mathscr{J}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) \quad = \quad & (\boldsymbol{q}^*, f') \\ & +\delta\mathscr{J}_{\partial\Omega}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) - BT_{\partial\Omega} \\ & +\delta\mathscr{J}_T(\boldsymbol{\phi};\delta\boldsymbol{\phi}) - BT_T \\ & -BT_0. \end{aligned} \tag{2.38}$$

This shows 4 parts that should not require the calculation of $\delta\boldsymbol{q}$.

The part on the right-hand side of (2.38), that integrates over the spatial domain at the boundary $t = 0$ is

$$-BT_0 = (\boldsymbol{u}^*, \delta\boldsymbol{u})_0 \,. \tag{2.39}$$

This term explicitly contains $\delta\boldsymbol{u}(\mathbf{x}, 0, \boldsymbol{\phi}, \delta\boldsymbol{\phi})$. If the initial velocity field is controlled, then $\delta\boldsymbol{u}(\mathbf{x}, 0, \boldsymbol{\phi}, \delta\boldsymbol{\phi})$ is a direct function of $\delta\boldsymbol{\phi}$ and the integral BT_0 can be evaluated without Navier–Stokes simulation. On the other hand, if the initial field is not controlled, $\delta\boldsymbol{u}(\mathbf{x}, 0, \boldsymbol{\phi}, \delta\boldsymbol{\phi})$, and BT_0 equal zero.

The Gateaux differential terms, on right-hand side of (2.38), integrated over the spatial domain at the time boundary $t = T$ are

$$\delta\mathscr{J}_T(\boldsymbol{\phi};\delta\boldsymbol{\phi}) - BT_T \quad = \quad \delta\mathscr{J}_T(\boldsymbol{\phi};\delta\boldsymbol{\phi}) - (\boldsymbol{u}^*, \delta\boldsymbol{u})_T \,. \tag{2.40}$$

To calculate this term without calculation of $\delta\boldsymbol{q}(\mathbf{x}, T, \boldsymbol{\phi}, \delta\boldsymbol{\phi})$, $u^*(\mathbf{x}, T, \boldsymbol{\phi})$ should be chosen such that the second term of (2.40) cancels the $\delta\boldsymbol{q}$ dependency of the first term, $\delta\mathscr{J}_T(\boldsymbol{\phi};\delta\boldsymbol{\phi})$. Since the second term contains only $\delta\boldsymbol{u}$, and no δp, it has to be possible to reformulate the Gateaux differential of $\mathscr{J}_T$ as $(\partial J_T(\boldsymbol{u}(\boldsymbol{\phi}))/\partial\boldsymbol{u}, \delta\boldsymbol{u})_T$. This imposes certain limitations on the choice of the cost functional. The 'initial' flow field of the adjoint at $t = T$ then has to be

$$u_i^*(\mathbf{x}, T, \boldsymbol{\phi}) = \frac{\partial J_T(\boldsymbol{q}(\boldsymbol{\phi}))}{\partial u_i}, \tag{2.41}$$

which is an analytical function of $\boldsymbol{q}(\mathbf{x}, T, \boldsymbol{\phi})$. Starting from $t = T$ the adjoint field is calculated back in time (see figure 2.5).

Finally, the Gateaux differential terms, on right-hand side of (2.38), that are integrated over the spatial boundaries remain. They correspond to

$$\begin{aligned}
&\delta\mathscr{J}_{\partial\Omega}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) - BT_{\partial\Omega} \\
= \;& \delta\mathscr{J}_{\partial\Omega}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) - \int_0^T \int_{\partial\Omega} \left((u_i' u_j + u_i u_j')u_i^* + p' u_j^* + u_j' p^*\right) n_j \mathrm{d}\mathbf{x}\mathrm{d}t \\
& - \int_0^T \int_{\partial\Omega} \frac{1}{Re}\left(-\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) u_i^* + u_i'\left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i}\right)\right) n_j \mathrm{d}\mathbf{x}\mathrm{d}t.
\end{aligned} \tag{2.42}$$

These terms are simplified significantly by taking the spatial boundary conditions of the non-linear and forward linearised equations into account. These boundary conditions partially prescribe $\boldsymbol{q}$, and $\delta\boldsymbol{q}$, and their derivatives on the spatial boundary $\partial\Omega$ such that several terms lose their $\delta\boldsymbol{q}$ dependence. The wish to calculate the remaining $\delta\boldsymbol{q}$-dependent terms of (2.42) without calculation of $\delta\boldsymbol{q}$, firstly imposes limitations on the formulation of the cost functional $\mathscr{J}_{\partial\Omega}$ and secondly determines the adjoint spatial boundary conditions.

Indeed, independent of the Navier–Stokes boundary conditions, equation (2.42) limits the cost functional $\mathscr{J}_{\partial\Omega}$ to cost functionals with Gateaux differential of the following shape:

$$\delta\mathscr{J}_{\partial\Omega}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) \;=\; \int_0^T \int_{\partial\Omega_j} a_i u_i' + b\, p' + c_i \left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) n_j \,\mathrm{d}\mathbf{x}\mathrm{d}t, \tag{2.43}$$

as these $\delta\boldsymbol{q}$ dependent terms are the only one represented in (2.42). a_i, b, and c_i are the coefficients of the Gateaux differential of the cost functional $\mathscr{J}_{\partial\Omega}$.

Using (2.42) and (2.43), the adjoint boundary conditions are derived in appendix A for two common boundary conditions: symmetry, and periodic boundary conditions. Also one common type of spatial boundary control is studied: control of the velocity at the boundary. During the derivations in the appendix A it is noticed that there may be an extra restriction on the formulation of (2.43), imposing an interrelationship between b, and c_i.

To conclude, if the adjoint boundary conditions fulfil the requirements described here and in appendix A, then the Gateaux differential of the total cost functional (2.25) is simplified to:

$$\delta\mathscr{J}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) = \left(\boldsymbol{f}', \boldsymbol{q}^*\right) + \left(\boldsymbol{u}^*, \delta\boldsymbol{u}\right)_0 + \left(f(\boldsymbol{q},\boldsymbol{q}^*), \delta\boldsymbol{q}\right)_{\partial\Omega}, \tag{2.44}$$

The Gateaux differential consists of three terms that each give the sensitivity to the control on a certain part of the domain. The three terms are now discussed in more detail.

The first term on the right-hand side of (2.44) is the contribution of the forcing $\boldsymbol{f}$ to the Gateaux differential. This term depends on the sensitivity of the forcing $\boldsymbol{f}'$ to a change $\delta\phi$. Only if the forcing is controlled, then $\boldsymbol{f}'$, and the first term, is non-zero. Hence, the contribution of the forcing $\boldsymbol{f}$ to the Gateaux differential $\left(\boldsymbol{f}', \boldsymbol{q}^*\right)$, is only non-zero if the forcing is controlled.

The second term on the right-hand side of (2.44) is the contribution from the control of the initial velocity field. Only if the initial velocity field is controlled, then the second term of (2.44) is non-zero ($\delta\boldsymbol{u}(\mathbf{x}, 0, \phi) = f(\boldsymbol{\phi})$).

The last term of (2.44) is only non-zero if some of the spatial boundary conditions are controlled (see appendix A) such that $\delta\boldsymbol{u}$ on the controlled wall is an explicit function of $\delta\boldsymbol{\phi}$. The last term is then $(f(\boldsymbol{q}, \boldsymbol{q}^*), g(\delta\boldsymbol{\phi}))_{\partial\Omega}$ with f and g an analytical function. The function g is zero in case the spatial boundary conditions are not controlled.

Figure 2.5 summarises the impact of the cost functional on the adjoint boundary conditions. The cost functional that measures a certain part of the field determines the adjoint boundary condition related to that field. For example the cost functional $\mathscr{J}_T$ determines the ‘initial’ adjoint field at time T.

Based on the Gateaux differential the gradient is determined (see, e.g. Protas, Bewley, and Hagen, 2004; Bewley et al., 2001). If we consider, for example, a control equal to the initial velocity field $\boldsymbol{u}(\mathbf{x}, 0, \phi) = \boldsymbol{\phi}$, then the gradient $\mathcal{G}$ is defined as:

$$\delta\mathscr{J}(\boldsymbol{\phi}; \delta\boldsymbol{\phi}) \quad = \quad (\mathcal{G}, \delta\boldsymbol{\phi})_0 \tag{2.45}$$

with the Gateaux differential $\delta\mathscr{J}(\boldsymbol{\phi}; \delta\boldsymbol{\phi}) = (\boldsymbol{u}^*, \delta\boldsymbol{\phi})_0$. This leads to the conclusion:

$$\mathcal{G} = \boldsymbol{u}^*(\mathbf{x}, 0, \phi) \tag{2.46}$$

2.5 Constrained optimisation

As discussed in section 2.1, in many optimisation and optimal control problems, constraints apply. The previous sections focussed on flow optimisation without constraints, this section presents several ways to adapt the unconstrained flow optimisation method to constrained flow optimisation.

The minimum of the cost functional is searched subject to constraints on the controls, called c_i (2.1). Examples of constraints c_i for flow optimisation are limitation of the energy consumption of the controls, continuity of the flow, and shape constraints in a design problem. The constraints in this thesis are equality constraints that relate directly to the controls and do not depend on the flow solution (see discussion in Giles and Pierce, 2000; Hinze et al., 2009). For flow optimisation in large parameters spaces, these simple constraints may become challenging.

Before we discuss the methods used in this work to impose the constraints, an important criterion for constrained optimisation methods is introduced: the Karush–Kuhn–Tucker (KKT) conditions (see e.g., Luenberger, 2003; Nocedal and Wright, 2006). These are necessary conditions for optimal points of a constrained optimisation problem. In case of convex problems, they are also sufficient conditions (Nocedal and Wright, 2006). The optimal (Lagrange) parameters are denoted with superscript $^\bullet$. The KKT conditions then state that for equality constrained problems there exists a Lagrange multiplier $\lambda^\bullet$, with components $\lambda_i^\bullet$, such that the following conditions are satisfied at $(\boldsymbol{\phi}^\bullet, \lambda^\bullet)$:

$$0 = \nabla_{\boldsymbol{\phi}} \mathscr{L}(\boldsymbol{\phi}^\bullet, \lambda^\bullet) = \nabla_{\boldsymbol{\phi}} \mathscr{J}(\boldsymbol{q}(\boldsymbol{\phi}^\bullet)) - \sum_{i \in I} \lambda_i^\bullet \nabla_{\boldsymbol{\phi}} c_i(\boldsymbol{\phi}^\bullet) \tag{2.47}$$

$$0 = c_i(\boldsymbol{\phi}^\bullet) \quad \text{for all } i \in I_1 \tag{2.48}$$

Condition (2.47) supposes that $\mathscr{J}$, and c_i are continuously differentiable at the optimum, and that the constraint gradients are linearly independent.

In this work the optimisation is subject to a linear and a non-linear constraint. For the linear constraint the parameter elimination method is used, while for the non-linear one we compare in the current manuscript two methods that formally ensure that an optimal point of the constrained optimisation problem is also a KKT point. These methods are the augmented Lagrangian method (see, e.g., Luenberger, 2003; Nocedal and Wright, 2006; Fletcher, 2000) and the gradient-projection method (see, e.g., Luenberger, 2003; Hinze et al., 2009). These three methods, i.e. parameter elimination, augmented Lagrangian, and gradient projection method, are presented next. The advantage of the augmented Lagrangian method over the less complex quadratic penalty method, is also discussed.

2.5.1 Augmented Lagrangian method

The augmented Lagrangian method (Luenberger, 2003; Nocedal and Wright, 2006), also called multiplier penalty function (Fletcher, 2000), is an advanced penalty method. First the augmented Lagrangian method is presented and subsequently the advantage of this method over the quadratic penalty method is detailed.

The augmented Lagrangian method inserts the constraint into the cost functional using a Lagrange term, and a quadratic penalty term (e.g. Nocedal and Wright, 2006):

$$\mathscr{L}_{\lambda,\mu}(\boldsymbol{\phi}) = \mathscr{J}(\boldsymbol{\phi}) - \lambda\, C(\boldsymbol{\phi}) + \frac{\mu}{2}\, C(\boldsymbol{\phi})^2. \tag{2.49}$$

with $C = [c_1, c_2, ..., c_{I_1}]^T$ the constraints $c_i = 0$ as defined in (2.1), λ the Lagrange multiplier, and μ the quadratic penalty weight. Sub problems $\mathscr{L}_{\lambda,\mu}$ are repeatedly optimised, using a method of choice (i.e., the non-linear CG method described in section 2.3). After convergence of each sub problem, the factors λ and μ are adapted (see Nocedal and Wright, 2006, algorithm 17.4 p.520), and more strict convergence criteria are selected, such that the optima of the sub problems gradually converge to a KKT point. If $\mu_i^{(n)}$ is large enough, the constraint violation is small and comparison of the gradient of (2.49) with (2.47) leads to the following approximation for $\lambda_i^\bullet$ (Nocedal and Wright, 2006):

$$\lambda_i^\bullet \approx \lambda_i^{(n)} - \mu_i^{(n)} c_i(\boldsymbol{\phi}^{(n)}). \tag{2.50}$$

This shows that when $\lambda_i^{(n)}$ approaches $\lambda_i^\bullet$, the constraint is less and less violated (see table 2.1). Therefore, if $\mu_i^{(n)}$ is large enough, the algorithm in Nocedal and Wright (2006) updates λ_i after each optimisation of $\mathscr{L}_{\lambda^{(n)},\mu^{(n)}}$ by setting $\lambda_i^{(n+1)}$ equal to the right hand side of (2.50): $\lambda_i^{(n+1)} = \lambda_i^{(n)} - \mu_i^{(n)} c_i(\boldsymbol{\phi}^{(n)})$. Subsequently $\mathscr{L}_{\lambda^{(n+1)},\mu^{(n+1)}}$ is optimised. The obvious advantage of this method is the ease of implementation with which constraints can be incorporated in an optimisation algorithm.

The difference between the augmented Lagrangian and quadratic penalty will now be outlined. The quadratic penalty method only adds the quadratic term to the cost functional:

$$\mathscr{L}_{\mu}(\boldsymbol{\phi}) = \mathscr{J}(\boldsymbol{\phi}) + \frac{\mu}{2}\, C(\boldsymbol{\phi})^2. \tag{2.51}$$

In order to satisfy the constraints, it is required that μ approaches infinity (cfr. table 2.1). This will lead to ill-conditioning of the system (Nocedal and Wright, 2006). In comparison the constraint violations are smaller for the augmented Lagrangian method (Nocedal and Wright, 2006, and table 2.1)

2.5.2 Gradient-projection method

The second method considered for optimisation subject to a non-linear constraint, is a gradient-projection method, which strictly projects the search line onto the surface described by the constraint (Luenberger, 2003) such that the constraint is satisfied at all steps during the optimisation process. If the constraint is linear,

Method	Quadratic penalty	Augmented Lagrangian
$c_i(\boldsymbol{\phi}^{(n)})$	$c_i(\boldsymbol{\phi}^{(n)}) \approx -\lambda_i^{\bullet}/\mu_i^{(n)}$	$c_i(\boldsymbol{\phi}^{(n)}) \approx -(\lambda_i^{\bullet} - \lambda_i^{(n)})/\mu_i^{(n)}$

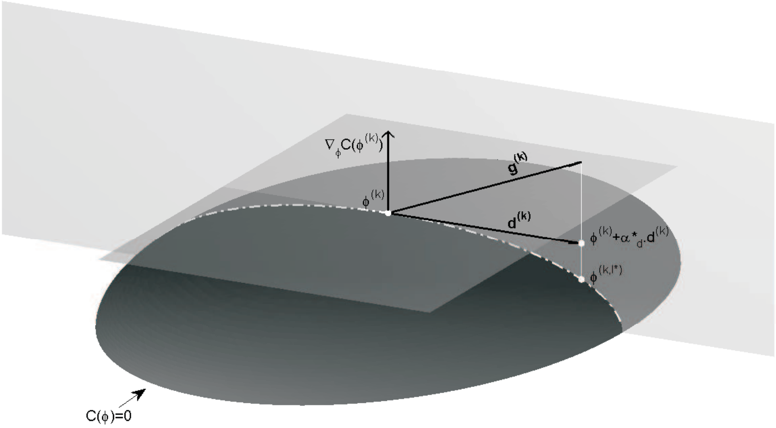
Table 2.1: Error on constraint $c_i(\boldsymbol{\phi}^{(n)})$ using quadratic penalty method and augmented Lagrangian method (Nocedal and Wright, 2006)

all parameters in the projected gradient direction are valid. If on the other hand the constraint is non-linear, after a step in the projected gradient direction, a correction has to be made to remain on the surface of parameters fulfilling the constraints.

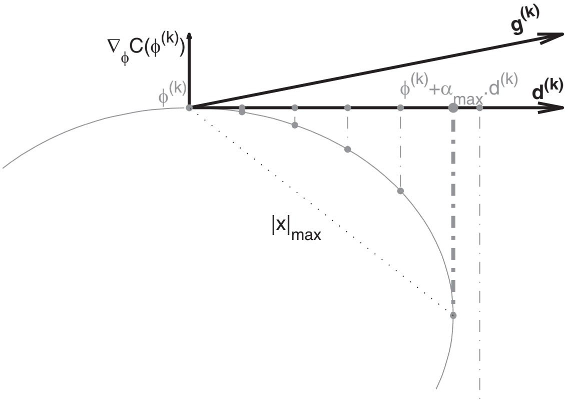
In Figure 2.6 (a), the principle is schematically illustrated in a three-dimensional parameter space for step k in the conjugate-gradient descent algorithm, which updates the parameters $\boldsymbol{\phi}^{(k)}$ to $\boldsymbol{\phi}^{(k+1)}$. The parameters $\boldsymbol{\phi}^{(k)}$ satisfy the non-linear constraint. The next point $\boldsymbol{\phi}^{(k+1)}$ is now obtained by finding the optimum in the gradient direction $\mathbf{g}^{(\mathbf{k})}$ projected on the surface $C(\boldsymbol{\phi}) = 0$ with C composed of rows of the constraints $C = [c_1, c_2, ..., c_{I_1}]^T$. To this end, the *Brent* algorithm is extended to take the non-linearity of the projected search direction into account. The standard *Brent* algorithm is an iterative algorithm (cf. Section 2.3.2), which requires cost-functional evaluations at several intermediate points $\boldsymbol{\phi}^{(k,l)}$, at a distance $x^{(k,l)}$, dictated by the algorithm, from the point $\boldsymbol{\phi}^{(k)}$ in the search direction (with $\mathrm{x}^{(k,l)} = \|\boldsymbol{\phi}^{(k)} - \boldsymbol{\phi}^{(k,l)}\|$). To find the point $\boldsymbol{\phi}^{(k,l)}$ on the projected gradient, we employ a two-level iterative algorithm. First of all, an estimate $\boldsymbol{\phi}^{(k,l^*)}$ is obtained as (Luenberger, 2003)

$$\boldsymbol{\phi}^{(k,l^*)} = (\boldsymbol{\phi}^{(k)} + \alpha_d^* \mathbf{d}^{(k)}) + \alpha_{\mathrm{proj}} \nabla_{\boldsymbol{\phi}} C(\boldsymbol{\phi}^{(k)}), \tag{2.52}$$

with $\mathbf{d}^{(\mathbf{k})}$ the projection of $\mathbf{g}^{(\mathbf{k})}$ on the surface tangent to $C(\boldsymbol{\phi})$ in the point $\boldsymbol{\phi}^{(k)}$ and the initial guess $\alpha_d^* = \mathrm{x}^{k,l}/\|\mathbf{d}^{(\mathbf{k})}\|$. The parameter α_{proj} is determined by using a linear null-point search algorithm such that $C(\boldsymbol{\phi}^{(k,l^*)}) = 0$ (Luenberger, 2003). Subsequently, in a set of iterations, α_d is adapted using a first-order search algorithm, such that $\|\boldsymbol{\phi}^{(k)} - \boldsymbol{\phi}^{(k,l^*)}\|$ converges to the distance $\mathrm{x}^{k,l}$; in a set of sub-iterations, α_{proj} is updated, such that the constraint in intermediate points remains satisfied. As shown in Figure 2.6 (b), it is possible for large distances $\mathrm{x}^{k,l}$, that no projected point $\boldsymbol{\phi}^{(k,l)}$ can be found. In that case x is reduced until a projected solution is found.



(a) 3D view



(b) 2D view with maximum distance

Figure 2.6: Gradient projection method

The projected gradient is calculated as

$$\mathbf{d}^{(k)} = \mathbf{g}^{(k)} - \nabla_{\boldsymbol{\phi}} C(\boldsymbol{\phi}^{(k)})^T [\nabla_{\boldsymbol{\phi}} C(\boldsymbol{\phi}^{(k)}) \nabla_{\boldsymbol{\phi}} C(\boldsymbol{\phi}^{(k)})^T]^{-1} \nabla_{\boldsymbol{\phi}} C(\boldsymbol{\phi}^{(k)}))\mathbf{g}^{(k)}, \tag{2.53}$$

The calculation of $\mathbf{d}^{(\mathbf{k})}$ requires inverting a matrix with size equal to the number of constraints. When the number of constraints is large, this can become prohibitive.

If during the optimisation procedure the magnitude of $\mathbf{d}^{(\mathbf{k})}$ vanishes and the conjugate gradient method falls back on the steepest descent method ($\beta = 0$ in (2.13)), an optimum is reached (Chong and Zak, 1996). In this case, the gradient of the cost functional is linearly dependent on the gradients of the constraints, implying that the cost functional can only further improve if a constraint is violated. The KKT conditions (2.47-2.48) are then satisfied (see Chong and Zak, 1996).

The projected gradient of the steepest descent direction is known to be a descent direction (Hinze et al., 2009; Brezillon and Gauger, 2004). In Hinze et al. (2009) it is however noted that this is not the case for the projection of any search direction, an example is given where the projection of the Newton search direction is not a descent direction (Hinze et al., 2009). In this work the search direction $\mathbf{g}^{(\mathbf{k})}$ is the conjugate-gradient direction. It is difficult to prove that the projection of the conjugate gradient is a descent direction, or $\mathbf{d}^{(\mathbf{k})}.\mathcal{G}^{(\mathbf{k})} < \mathbf{0}$ with $\mathbf{g}^{(\mathbf{k})}$ in (2.53) given by (2.13). Equation (2.13) however shows that if the step lengths decrease, for example near the optimum, the gradient $\mathcal{G}^{(k-1)}$ and $\mathcal{G}^{(k)}$ become equal, and β goes to zero. The conjugate gradient method is then reduced to the steepest descent direction for which it is known that the projection is still a descent direction. If β is non negligible and the projection would result in an ascent direction, this is not a problem as the line-search algorithm *mnbrack* and *Brent* (Press et al., 1996, and section 2.3.2) are robust enough to handle this situation. In that case, the line search searches for negative step lengths.

The conjugate gradient method combined with the gradient projection method shows similarities with line search algorithms on manifolds (see, e.g., Absil, Mahony, and Sepulchre, 2008) where the parameter surface is related to the manifold, and the α_{proj} correction step to the retraction. Optimisation on manifolds is used in the framework of matrix optimisation problems (Absil et al., 2008; Vandereycken and Vandewalle, 2010).

2.5.3 Parameter-elimination method

The last method used is the parameter-elimination method, which reduces the parameter set to a parameter set that can be freely optimised without constraints. It is a very convenient method for linear constraints and might for some cases also be used for non linear constraints (see, for more details Nocedal and Wright, 2006). The method reduces the parameters to a subset of linear independent parameters. Instead of formulating the iterative optimisation algorithm (cf. section 2.3) in terms of $\boldsymbol{\phi}$, it is directly formulated using the reduced parameters. The application of the parameter-elimination method to the envisaged optimisation problem is discussed in section 3.3.1.

Chapter 3

Numerical approach

This chapter starts with the specification of the optimisation problem for the temporal mixing layer (section 3.1). Subsequently the computational set-up, discretisation of the flow, and adjoint simulations are discussed in section 3.2. Finally, the application of the constraint methods on the envisaged optimisation problem is detailed in section 3.3.

3.1 Optimisation problem

In Chapter 2 the flow optimisation techniques were introduced for a generic cost functional and constraints on the parameters. Here, the techniques are applied to the temporal mixing layer, introduced in section 1.2.

The perturbations $\boldsymbol{\phi}(\mathbf{x})$ on the initial mean flow field of the temporal mixing layer are optimised such that an optimal volume-averaged property is achieved after a selected simulation time, called time horizon T. For the initial mean velocity field a hyperbolic-tangent in the streamwise direction is taken. The dimensionless initial velocity perturbations $\boldsymbol{\phi}$, which have no mean velocity component, are added to this dimensionless mean velocity field (see figure 3.1):

$$\boldsymbol{u}(\mathbf{x}, 0, \boldsymbol{\phi}) = \tanh(x_3)\, \mathbf{e}_1 + \boldsymbol{\phi}(\mathbf{x}), \tag{3.1}$$

with $\mathbf{e}_1$ the unit vector in the streamwise direction x_1 (see figure 3.1). This velocity field was made dimensionless according to section 2.1.1 with as reference velocity $u_{ref} = \Delta U/2\ [m/s]$, and as reference length $L_{ref} = \delta_\omega/2\ [m]$, with ΔU the velocity

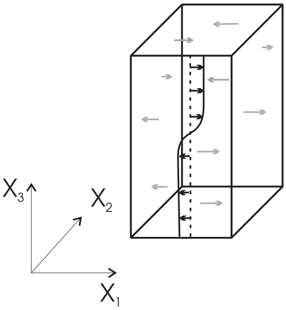


Figure 3.1: Schematic illustration of a temporal mixing layer. Mean velocity profile (—) and the perturbations (grey arrows).

difference between the upper and lower flow, and δ_ω the initial vorticity thickness:

$$\delta_\omega = \frac{\Delta U}{\left.\left(\dfrac{\partial\langle \boldsymbol{u}(\mathbf{x},0,\boldsymbol{\phi})\rangle}{\partial x_3}\right)\right|_{max}}. \tag{3.2}$$

The ensuing evolution of the flow is governed by the non-dimensional incompressible Navier–Stokes equations (2.11-2.12).

The temporal mixing-layer case is used as a substitute case to study how long flow control can affect the evolution of the mixing-layer solution. Several cost functionals are formulated that measure volume-averaged properties at the time horizon T, such that $\mathscr{J}$ (Eq. (2.14)) simplifies to $\mathscr{J}_T$.

The optimisation will minimise the cost functional subject to two constraints on $\boldsymbol{\phi}(\mathbf{x})$, i.e.

$$\min_{\boldsymbol{\phi}} \mathscr{J}_T(\boldsymbol{q}(\mathbf{x},T,\boldsymbol{\phi})) \quad \text{subject to} \quad \begin{cases} \nabla\cdot\boldsymbol{\phi}(\mathbf{x}) = 0 & \forall \mathbf{x}\in\Omega \quad (3.3) \\ \dfrac{1}{2}\dfrac{1}{\Omega}\displaystyle\int_\Omega \boldsymbol{\phi}\cdot\boldsymbol{\phi}\,\mathrm{d}\mathbf{x} = E_0\dfrac{1}{2} & (3.4) \end{cases}$$

The first constraint (3.3) follows from the requirement that the initial velocity field $\boldsymbol{u}(\mathbf{x},0,\boldsymbol{\phi})$ should satisfy the continuity equation. When the system is discretised, this constraint has to be satisfied in every point on the grid. This leads to an equation per grid point expressing the continuity of the discretised $\boldsymbol{\phi}$. The enforcement of these constraints is essential, since it is required for a physically relevant solution of the Navier–Stokes equations (2.11–2.12).

The second constraint (3.4) keeps the dimensionless total energy per unit volume of the perturbations fixed at a constant level E_0 times the initial dimensionless mean-flow energy per unit volume (1/2 for a box infinite in the normal direction). From a practical point of view, the kinetic energy of the controls (i.e. the initial perturbations of the temporal mixing layer) should be low compared to the mean-flow energy, since, if we would consider an actual implementation of the control in a spatial framework, this energy would be roughly related to the amount of work required from an upstream actuator array on a splitter plate. However, in the current work, it is not the intention to elaborate on the connection between the optimised perturbations in the temporal framework and the actuations in the spatial framework. It is obvious that controls, i.e. the perturbations, with a high level of energy can have a large impact on the mean-flow statistics; they correspond to a brute-force approach to the mixing layer flow control problem. The effectiveness of low-energy controls on the mean-flow evolution over long time horizons seems less obvious. Therefore, this work focusses on low-energy controls, and strictly enforces a selected constant energy level of the controls as a constraint during the optimisation.

3.2 Flow and adjoint simulation

3.2.1 Computational setup

Navier–Stokes simulation

We consider a temporal mixing layer with an initial Reynolds number equal to 50 (based on half the velocity difference ($\Delta U/2$), and half the initial vorticity thickness ($\delta_\omega/2$)). This corresponds to the Reynolds number of the mixing layer studied by Vreman et al. (1997). The mixing layer is simulated with incompressible DNS.

Periodic boundary conditions are used both in streamwise (x_1) and spanwise (x_2) directions (the directions of the axes are indicated in figure 3.1). In the normal direction (x_3), the boundaries are located at $x_3 = \pm L_3/2$ with L_3 the dimensionless length of the box in the normal direction. Symmetry boundary conditions are imposed on these boundaries, which are sufficiently far from the mixing region around $x_3 = 0$ to exclude interactions between the boundary and the mixing region.

The initial flow field is given by equation (3.1). It is well documented that for perturbations with sufficiently large wave lengths in the streamwise direction, the mixing layer is unconditionally unstable (see e.g. Michalke, 1964; Monkewitz and Huerre, 1982). We select a streamwise domain size $L_1 = n\lambda_1$, with λ_1 the most unstable wavelength following from linear stability theory. For a tanh-profile, and $Re = 50$, λ_1 is 15.4. For the spanwise direction, we take $L_2 = n\lambda_2$, with $\lambda_2 = 0.6\lambda_1$. λ_2 corresponds to the wavelength of the most unstable spanwise perturbation on the rollers originating from the Kelvin–Helmholtz instability in the streamwise direction (Pierrehumbert and Widnall, 1982). n is set to 4 during the verification tests in chapter 4 and $n = 8$ for the simulations presented in chapter 5 and 6.

Adjoint simulation

The adjoint simulation is performed in the same domain as the Navier–Stokes simulations and uses the same Reynolds number. The adjoint boundary conditions are defined by the derivation in section 2.4.1 and appendix A, taking into account that $\mathscr{J} = \mathscr{J}_T$ for the current optimisation.

The adjoint spatial boundary conditions in the stream and spanwise direction are periodic corresponding with the forward simulation (see appendix section A.2). For the normal direction, the equations (A.11-A.13) have to be met. This is the case if the adjoint boundary conditions are symmetry and impermeability on the top and bottom plane (see appendix section A.1) like in the forward problem.

The initial condition for the adjoint field $\boldsymbol{u}^*$, which is given at time T (the characteristics of the adjoint problem propagate back in time), is given by equation (2.41): $\boldsymbol{u}^*(T) = \partial J_T(\boldsymbol{u}(\boldsymbol{\phi}))/\partial \boldsymbol{u}$. Since the cost functional does not integrate over time ($\mathscr{J}_\Omega = 0$), according to equation (2.24): $\boldsymbol{f}^* = 0$.

The Gateaux differential of the cost functionals (2.25) is then reduced to:

$$\delta \mathscr{J}(\boldsymbol{\phi}; \delta\boldsymbol{\phi}) = \int_\Omega \boldsymbol{u}^*(\mathbf{x}, 0, \boldsymbol{\phi}).\delta\boldsymbol{u}(\mathbf{x}, 0, \boldsymbol{\phi}) \, \mathrm{d}\mathbf{x}. \tag{3.5}$$

3.2.2 Navier–Stokes and adjoint discretisation

For the direct numerical simulations of the Navier–Stokes equations (2.11,2.12), a mixed pseudo-spectral finite-difference code is employed (Meyers and Sagaut, 2007). The code is parallellised with Open MPI (Delport et al., 2009). This section first presents the numerical techniques employed in the Navier–Stokes code. The pseudo-spectral discretisation is discussed, followed by the finite-difference discretisation, subsequently the Poisson solver and time integration is addressed. At the end of the section the differences between the adjoint code and Navier–Stokes code are outlined.

Pseudo-spectral Fourier discretisation

In the two periodic directions (stream and spanwise) a pseudo-spectral Fourier discretisation is used. Pseudo-spectral implies that some of the operations are performed in the spectral domain and others in the physical domain. First spectral discretisation is outlined and subsequently the advantage of the pseudo-spectral method over the spectral method is discussed.

In case of spectral discretisation the field is represented as truncated Fourier series, for a $N \times N$ grid:

$$\begin{aligned}
&\boldsymbol{u}(\mathbf{x}) \\
&= \sum_{\frac{k_1 L_1}{2\pi}=0}^{(\frac{N}{2}-1)} \sum_{\frac{k_2 L_2}{2\pi}=0}^{(\frac{N}{2}-1)} a(x_3)\cos(k_1 x_1 + k_2 x_2) - b(x_3)\sin(k_1 x_1 + k_2 x_2) \\
&= \sum_{\frac{k_1 L_1}{2\pi}=0}^{(\frac{N}{2}-1)} \sum_{\frac{k_2 L_2}{2\pi}=0}^{(\frac{N}{2}-1)} \frac{a + \imath b}{2}(x_3) e^{\imath(k_1 x_1 + k_2 x_2)} + \frac{a - \imath b}{2}(x_3) e^{-\imath(k_1 x_1 + k_2 x_2)} \\
&= \sum_{\frac{k_1 L_1}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \sum_{\frac{k_2 L_2}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \hat{\boldsymbol{u}}_{(k_1,k_2)}(x_3) e^{\imath(k_1 x_1 + k_2 x_2)}
\end{aligned} \tag{3.6}$$

where $\hat{\boldsymbol{u}}_{(k_1,k_2)}$ and $\hat{\boldsymbol{u}}_{-(k_1,k_2)}$ are complex conjugates. The Hermitian symmetry of the Fourier coefficients $\hat{\boldsymbol{u}}$ allows to store only the Fourier coefficients of positive wave numbers (see figure 3.2). The hat symbol ($\hat{\mathbf{a}}$) denotes the Fourier coefficients of $\mathbf{a}$.

The advantage of spectral discretisation is that for infinitely differentiable functions with periodic boundary conditions, its convergence in function of the number of grid points N is faster than any finite power of $1/N$ (Canuto et al., 2006). The faster convergence is related to the method used to calculate derivatives. Derivatives in the spectral directions are calculated in spectral space by multiplication of each $\hat{\boldsymbol{u}}_{\mathbf{k}}$ with $\imath$ times the wavenumber in the derivative direction. This derivative calculation is more accurate than subtracting velocities on neighbouring locations as used in the finite difference method.

The spectral method has, however, two important disadvantages. First of all, the boundary conditions have to be periodic. Secondly, the convolution operation to calculate $u_i u_j$ in the spectral domain is more expensive ($\mathcal{O}(N^2)$ operations per spectral direction) than the product $u_i u_j$ in the physical domain ($\mathcal{O}(N)$ operations per direction) (see, e.g., Canuto et al., 2006).

The enhanced spectral method, the pseudo-spectral method (see, e.g., Canuto et al., 2006, and used in this work), calculates the derivatives in the spectral directions in the spectral domain to take advantage of the accuracy of the spectral method, but reduces the computational cost of convolutions (for the calculation of $u_i u_j$) by computing $u_i u_j$ in physical space. When the field is transformed from

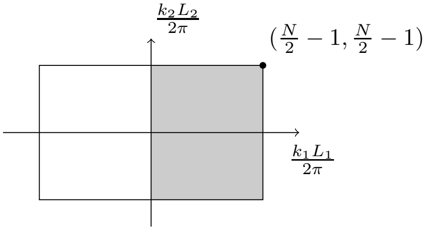


Figure 3.2: Wave numbers occurring in plane (x_1, x_2) on grid $N \times N$. Complex velocities in white and grey rectangle are complex conjugates. Complex velocities in grey rectangle are stored.

the spectral to the physical domain and back after multiplication, the cost of the convolution is $\mathcal{O}(N \log_2(3/2N))$ operations per direction including dealiasing of the non-linear terms using the 3/2 dealiasing rule (Canuto et al., 2006), which is a significant reduction compared to $\mathcal{O}(N^2)$ operations.

Dealiasing of the non linear term (Canuto et al., 2006) is required as the convective term $(u_i u_j)$ has wave numbers with values up to twice the maximum wave numbers represented on the grid. The frequencies that cannot be represented on the grid have to be cut off, otherwise they introduce errors on the solution at other frequencies (aliasing). To this end the 3/2 dealiasing rule is generally used (see, e.g., Canuto et al., 2006), which is outlined in the remainder of this section.

Dealiasing prevents waves with wavenumber $k_1 \mp N\ k_{1,min}$ (with $k_{1,min} = 2\pi/L_1$) to introduce errors on the waves with wavenumber k_1 when a grid with N points and equidistant grid spacing Δx is used, i.e.:

$$\cos((k_1 + N k_{1,min}) j \Delta x) = \cos(k_1 j \Delta x + j\ 2\pi) = \cos(k_1 j \Delta x) \tag{3.7}$$

In the 3/2 dealiasing rule the non linear terms are calculated on a $3N/2$ grid. The $\hat{\boldsymbol{u}}_{(k_1,k_2)}$ coefficients are copied to a $3N/2$ grid and the additional wave numbers are padded with zeros. The convolutions $u_i u_j$ are performed and still lead to wave numbers that are not representable on the 3/2 grid. It are the wave numbers $\mp(3N/4, ..., N) k_{1,min}$ (area between middle and largest rectangle in fig. 3.3), the corresponding waves impact the solution at a wave distance $3N/2\ \ k_{1,min}$ on the $3N/2$ grid (in analogy with equation 3.7). The waves $3N/4\ \ k_{1,min}$ and $N\ \ k_{1,min}$, for example, introduce respectively an error on the waves $3N/4\ k_{1,min} - 3N/2\ k_{1,min} = -3N/4\ k_{1,min}$ and $N\ k_{1,min} - 3N/2\ k_{1,min} = -N/2\ \ k_{1,min}$. To conclude, the impacted waves are $(\mp((N/2, ..., 3N/4)\ \ k_{1,min})$ (area between inner and middle rectangle in fig. 3.3) and are removed when the spectral field is truncated to the maximum wavenumber $(N/2-1)\ k_{1,min}$ of the N grid.

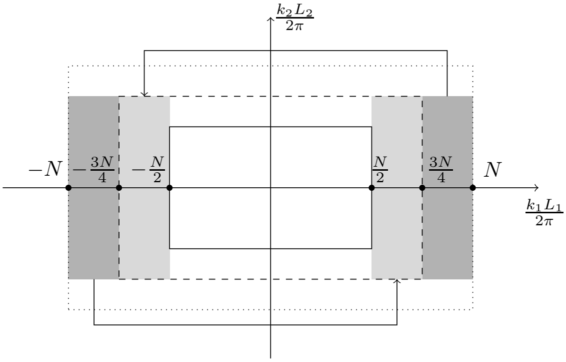


Figure 3.3: Wave numbers occurring in plane (x_1, x_2) on grid N (inside –), grid $3N/2$ (inside $--$) and in non-linear terms (inside $\cdots$). Wave numbers in dark grey zone cause aliasing errors on light grey zone. The field mapped from grid $3N/2$ on grid N is de-aliased

Finite-difference discretisation

The normal direction is discretised using a fourth-order energy-conserving staggered finite-difference discretisation (see Verstappen and Veldman, 2003), and the locations of normal velocities u_3 are shifted in the normal direction by half a cell compared to p, u_1, and u_2. In order to conserve mass, momentum, and energy (in absence of physical dissipation), the convection is approximated with a skew-symmetric discrete operator in accordance with the properties of the differential operator (Verstappen and Veldman, 2003). The skew-symmetry is only respected if the weights are independent of the grid non-uniformity. This differs from the usual fourth order method, where the weights are adapted to the grid to obtain the local fourth order convergence. Verstappen and Veldman (2003) show that nevertheless the energy conservative scheme is fourth order accurate on a non-uniform mesh.

An explicit Runge–Kutta time integration is used with direct solve of the pressure through the Poisson equation to retain continuity. The Poisson equation solver is discussed first, subsequently the Runge–Kutta time integration is shown.

Poisson solver

The Poisson equation (3.8) stems from the divergence of the momentum equation (2.12):

$$\frac{\partial^2 u_i u_j}{\partial x_i \partial x_j} + \nabla^2 p = 0 \tag{3.8}$$

The first term of this equation is written as truncated Fourier series:

$$\frac{\partial^2 u_i u_j}{\partial x_i \partial x_j} = \sum_{\frac{k_1 L_1}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \sum_{\frac{k_2 L_2}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \hat{c}_{(k_1,k_2)}(x_3) e^{\imath(k_1 x_1 + k_2 x_2)} \tag{3.9}$$

The weak Galerkin formulation requires that:

$$\int_\Omega \left(\frac{\partial^2 u_i u_j}{\partial x_i \partial x_j} + \nabla^2 p \right) \psi_{(r_1,r_2)} \, \mathrm{d}\mathbf{x} = 0 \tag{3.10}$$

where $\psi_{(r_1,r_2)}(\mathbf{x}) = \exp(\imath(r_1 x_1 + r_2 x_2))$ is a test function. Equation 3.10 has to be fulfilled for every test function $\psi_{(r_1,r_2)}$ with $r_m \in (2\pi/L_m)[-(N/2-1), ..., (N/2-1)]$. Equations (3.9) and (3.10) lead to the requirement:

$$\begin{aligned}
&\int_\Omega \sum_{\frac{k_1 L_1}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \sum_{\frac{k_2 L_2}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \left(\hat{c}_{(k_1,k_2)}(x_3) e^{\imath(k_1 x_1 + k_2 x_2)} \right) e^{\imath(r_1 x_1 + r_2 x_2)} \, \mathrm{d}\mathbf{x} \\
&+ \int_\Omega \sum_{\frac{k_1 L_1}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \sum_{\frac{k_2 L_2}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \left(\frac{\partial^2}{\partial x_i^2} \hat{p}_{(k_1,k_2)}(x_3) e^{\imath(k_1 x_1 + k_2 x_2)} \right) e^{\imath(r_1 x_1 + r_2 x_2)} \, \mathrm{d}\mathbf{x} \\
&= 0 \\
&\qquad \forall r_m \in \frac{2\pi}{L_m}[-(\frac{N}{2}-1), ..., (\frac{N}{2}-1)] \text{ with } m \in [1,2]
\end{aligned} \tag{3.11}$$

Analytical integration of (3.11) leads to an equation per wavenumber (k_1, k_2):

$$\left(-k_1^2 - k_2^2 + \mathbf{DD_3}\right) \hat{p}_{(k_1,k_2)}(x_3) \quad = \quad -\hat{c}_{(k_1,k_2)}(x_3) \tag{3.12}$$

with $\mathbf{DD_3}$ the matrix representing the finite-difference discretisation of ∂^2/x_3^2. The equations are decoupled per wavenumber, leading to $N_1/2 \times N_2$ different sets of N_3 equations (factor 2 because $\hat{p}_{(k_1,k_2)}$ and $\hat{p}_{-(k_1,k_2)}$ are complex conjugates). Each set of N_3 equations is described by a sparse matrix system with bandwidth 7. The system is solved using a direct solver based on LU-decomposition.

Time integration

The time integration is performed by a classical four-stage fourth-order accurate Runge–Kutta time integration (see, e.g. Ferziger and Peric, 2002):

$$\boldsymbol{u}(t^*_{n+\frac{1}{2}}) = \boldsymbol{u}(t_n) + \frac{\Delta t}{2} RHS(\boldsymbol{q}(t_n)) \tag{3.13}$$

$$\boldsymbol{u}(t^{**}_{n+\frac{1}{2}}) = \boldsymbol{u}(t_n) + \frac{\Delta t}{2} RHS(\boldsymbol{q}(t^*_{n+\frac{1}{2}})) \tag{3.14}$$

$$\boldsymbol{u}(t^*_{n+1}) = \boldsymbol{u}(t_n) + \Delta t \; RHS(\boldsymbol{q}(t^{**}_{n+\frac{1}{2}})) \tag{3.15}$$

$$\begin{aligned}\boldsymbol{u}(t_{n+1}) = \boldsymbol{u}(t_n) &+ \frac{\Delta t}{6}\big(RHS(\boldsymbol{q}(t_n)) + 2RHS(\boldsymbol{q}(t^*_{n+\frac{1}{2}})) \\ &+2RHS(\boldsymbol{q}(t^{**}_{n+\frac{1}{2}})) + RHS(\boldsymbol{q}(t^*_{n+1}))\big)\end{aligned} \tag{3.16}$$

with $RHS(\boldsymbol{q})$ equal to $-\boldsymbol{u}\cdot\nabla\boldsymbol{u} - \nabla p + \frac{1}{Re}\nabla^2\boldsymbol{u} + f$ (cfr. (2.12)) that is discretised with the techniques described before. Time steps Δt are restricted by setting both the convective and diffusive Courant–Friedrichs–Lewy number (CFL number):

$$CFL_{conv} = \frac{\max(\langle u_1\rangle)\Delta t}{\Delta x_1} \tag{3.17}$$

$$CFL_{diff} = \frac{\nu \Delta t}{\min(\Delta x_3)^2} \tag{3.18}$$

Discretisation and time integration for the adjoint problem

To discretise the adjoint equations (2.31, 2.32), the same numerical schemes for the spatial derivatives are used as for the Navier–Stokes equations (i.e. pseudo-spectral in x_1, x_2, and finite-volume in x_3).

The diffusive terms of the adjoint equations have the same shape as the diffusive terms of the Navier–Stokes equations, and are discretised in exactly the same way. The adjoint convective terms on the other hand differ to a large extent from the Navier–Stokes convective terms (see table 3.1). The adjoint convective term requires the calculation of twice as many product terms than the normal convective term ($u_j\partial u_j^*/\partial x_i$,$u_j\partial u_i^*/\partial x_j$ compared with u_iu_j). To calculate these terms, several variable fields have to be transformed from complex to real space. Table 3.1 shows that 3 and 12 variables are Fast Fourier transformed from complex to real space, respectively for the normal and adjoint convective term. The convolutions are calculated and transformed back to the complex space. For the normal, and

	Navier–Stokes solver	Adjoint solver
Convective term contributing to $\partial u_i/\partial t$	$\sum_{j=1}^{3} \frac{\partial u_i u_j}{\partial x_j}$	$\sum_{j=1}^{3} u_j \frac{\partial u_j^*}{\partial x_i} + u_j \frac{\partial u_i^*}{\partial x_j}$
Product in real space	$u_i u_j$	$u_j \frac{\partial u_j^*}{\partial x_i}$, $u_j \frac{\partial u_i^*}{\partial x_j}$
FFT from complex to real	u_i (3)	u_i (3), $\frac{\partial u_j^*}{\partial x_i}$ (9)
FFT from real to complex	$u_1 u_1$ (1), $u_3 u_1$ (1), $u_i u_2$ (3), $\frac{\partial u_i u_3}{\partial x_3}$ (3)	$\sum_{j=1}^{3} u_j \frac{\partial u_j^*}{\partial x_i} + u_j \frac{\partial u_i^*}{\partial x_j}$ (3)
number of FFT	$3 + 8$	$12 + 3$

Table 3.1: Computational cost of the convective terms in the Navier–Stokes solver and the adjoint Navier–Stokes solver

the adjoint convective term there are respectively 8 and 3 convolutions that have to be transformed back separately. The 8 convolutions stem from the wish to derive $u_1 u_1$, $u_3 u_1$, and $u_i u_2$ to x_1 and x_2 in spectral space to take advantage of the accuracy of the spectral discretisation. In total, the number of FFT transforms during the calculation of the adjoint convective term is 36% higher than for the normal convective term. Although the fast Fourier transforms are one of the most time consuming parts of the code, the calculation time for an adjoint simulation and a DNS simulation are comparable with the current code. This is probably related to the impact of the MPI communication cost.

For the time integration, a four-stage fourth-order Runge–Kutta method is used like in the Navier–Stokes solve (cfr. (3.13-3.16)) but now backwards in time:

$$\boldsymbol{u}(t^*_{n-\frac{1}{2}}) = \boldsymbol{u}(t_n) - \frac{\Delta t}{2}\, aRHS(\boldsymbol{u}(t_n), \boldsymbol{q}^*(t_n)) \tag{3.19}$$

$$\boldsymbol{u}(t^{**}_{n-\frac{1}{2}}) = \boldsymbol{u}(t_n) - \frac{\Delta t}{2}\, aRHS(\boldsymbol{u}(t_n), \boldsymbol{q}(t^*_{n-\frac{1}{2}})) \tag{3.20}$$

$$\boldsymbol{u}(t^*_{n-1}) = \boldsymbol{u}(t_n) - \Delta t\, aRHS(\boldsymbol{u}(t_n), \boldsymbol{q}(t^{**}_{n-\frac{1}{2}})) \tag{3.21}$$

$$\begin{aligned} \boldsymbol{u}(t_{n-1}) = \boldsymbol{u}(t_n) - \frac{\Delta t}{6}\Big(& aRHS(\boldsymbol{u}(t_n), \boldsymbol{q}(t_n)) + 2\, aRHS(\boldsymbol{u}(t_n), \boldsymbol{q}(t^*_{n-\frac{1}{2}})) \\ & + 2\, aRHS(\boldsymbol{u}(t_n), \boldsymbol{q}(t^{**}_{n-\frac{1}{2}})) + aRHS(\boldsymbol{u}(t_n), \boldsymbol{q}(t^*_{n-1}))\Big) \end{aligned} \tag{3.22}$$

with $aRHS(\boldsymbol{u}, \boldsymbol{q}^*) = \boldsymbol{u} \cdot \nabla \boldsymbol{u}^* + \boldsymbol{u} \cdot (\nabla \boldsymbol{u}^*)^T + (1/Re)\nabla^2 \boldsymbol{u}^* + \nabla p^*$

The adjoint flow is solved at the same points in time, with the same Δt, as in the accompanying non-linear simulations because to solve the adjoint equations in a point $\boldsymbol{\phi}$, the solution of the preceding Navier–Stokes equations $\boldsymbol{u}(\mathbf{x}, t_n, \boldsymbol{\phi})$ at this point is required (cf. (2.23, 2.32), fig. 2.5, and fig. 3.4). This solution is stored onto disk during the preceding simulation. To limit required disk storage, and disk access time, we select to store this solution only at every time step, and not at intermediate stages in the Runge–Kutta iteration. At the beginning of every Runge–Kutta time step of the adjoint system, the non-linear solution $\boldsymbol{u}(\mathbf{x}, t_n, \boldsymbol{\phi})$ at step t_n is read from the disk (see figure 3.4), but subsequently kept constant during the Runge–Kutta stages (i.e. $\boldsymbol{u}(\mathbf{x}, t_n, \boldsymbol{\phi})$ is used for all calculations of $aRHS$ in the predictor and corrector steps (3.19-3.22)). Compared to a discrete adjoint approach (where the Navier–Stokes equations are first discretised, before the linearised adjoint system is derived based on this discrete system), this formulation reduces the required disk storage by a factor of 4.

If necessary, there are methods to reduce the required disk space even more. One method is to save the flow solution only at 1 out of 2 grid points and every n time steps (Bewley et al., 2001; Wei and Freund, 2006) and to use interpolation in time and space to obtain $\boldsymbol{u}(\mathbf{x}, t_n, \boldsymbol{\phi})$ at every point, and every time step. Another method is called checkpointing (Griewank and Walther, 2000; Hinze and Sternberg, 2005), and reduces the storage need by storing the flow solution only at a limited amount of time steps, the checkpoints, and by recalculating the flow solution when needed: when the adjoint simulation reaches a checkpoint t_2, the forward simulation is repeated starting from the checkpoint earlier in time t_1 (the adjoint simulation is integrated back in time) to the current checkpoint t_2, and the flow solution is stored at every time step. Subsequently the adjoint simulation is integrated in time from t_2 to t_1. This process is repeated until $t = 0$ is reached. For the optimisation problems described in this work, it is feasible to store the flow solution at every time step, so storing only a part of the solution, and using interpolation or checkpointing was not considered.

A drawback, which is well known for the continuous adjoint methodology, is that the gradient obtained from the discretised adjoint equations, is not fully consistent with the gradient of the discretised Navier–Stokes system (cf. 2.4). In our case, due to the piecewise constant approximation of $\boldsymbol{u}(\mathbf{x}, t, \boldsymbol{\phi})$ during the Runge–Kutta iteration of the adjoint equations, this inconsistency in the gradient is first-order in the time step. This ‘error’ on the gradient will be assessed in more detail in section 4.1.

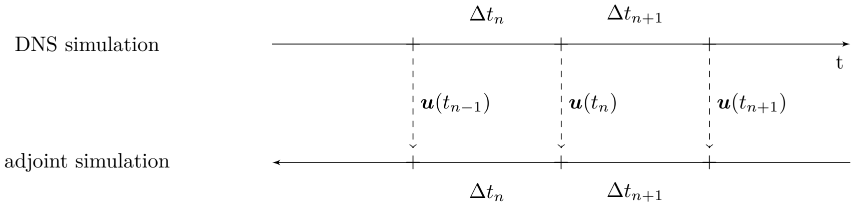


Figure 3.4: The adjoint flow is solved at the same points in time, with the same Δt, as the Navier-Stokes flow. At the beginning of every Runge–Kutta time step of the adjoint system, the non-linear solution $\boldsymbol{u}(\mathbf{x}, t_n, \boldsymbol{\phi})$ at step t_n is read from the disk

3.3 Optimisation in the discretised parameter space

In chapter 2, techniques for optimisation with constraints on the control were presented. In section 3.3.1 these techniques are applied on the mixing-layer optimisation problem, which is subject to the constraints (3.3) and (3.4).

Optimisation in the discretised parameter space showed the need to regularise the gradient. The regularisation method is discussed in section 3.3.2.

3.3.1 Constraints

The current optimisation problem has two constraints, i.e. (3.3) and (3.4). Both constraints are enforced on the discretised representation $\boldsymbol{\Phi}$ of the controls $\boldsymbol{\phi}$. First the technique used for the continuity constraint is presented, subsequently the energy constraint is elaborated on.

Continuity constraint

The first constraint (3.3) ensures continuity of the initial condition $\boldsymbol{u}(\mathbf{x}, 0, \boldsymbol{\phi})$. According to the weak Galerkin formulation and using the discretised representation of the ∇-operator, continuity requires:

$$\begin{aligned} 0 &= \int_\Omega \left(\frac{\partial u_i}{\partial x_i} \right) \psi_{(r_1,r_2)}\, \mathrm{d}\mathbf{x} \qquad (3.23) \\ &= \int_\Omega \sum_{\frac{k_1 L_1}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \sum_{\frac{k_2 L_2}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \big(\imath k_1 \hat{u}_{1,(k_1,k_2)} + \imath k_2 \hat{u}_{2,(k_1,k_2)} + \mathbf{D_z} \hat{u}_{3,(k_1,k_2)} \big) \\ &\qquad\qquad e^{\imath(k_1x_1+k_2x_2)} \cdot \hat{a}\; e^{\imath(r_1x_1+r_2x_2)}\, \mathrm{d}\mathbf{x} \\ &\qquad \forall r_m \in \frac{2\pi}{L_m}[-(\frac{N}{2}-1),...,(\frac{N}{2}-1)] \qquad (3.24) \end{aligned}$$

with $\imath$ the imaginary unit, and $\mathbf{D_z}$ the matrix representing the finite-difference discretisation of $\partial/\partial x_3$ in the normal direction and $\psi_{(r_1,r_2)}$ a test function. Analytical integration leads to the continuity equation per mode (k_1, k_2):

$$\imath k_1 \hat{u}_{1,(k_1,k_2)}(x_3) + \imath k_2 \hat{u}_{2,(k_1,k_2)}(x_3) + \mathbf{D_z} \hat{u}_{3,(k_1,k_2)}(x_3) = 0. \qquad (3.25)$$

Note that the continuity equations are also decoupled per mode. This implies that the continuity of the initial velocity field is ensured if $\mathbf{\Phi}_{(k_1,k_2)}$ is divergence free for each mode (k_1, k_2) it contains. This is a linear constraint that has to be fulfilled by the optimisation parameters.

The constraint (3.25) is straightforwardly enforced through parameter elimination during the optimisation. The discretised controls $\mathbf{\Phi} = [\mathbf{\Phi}_1, \mathbf{\Phi}_2, \mathbf{\Phi}_3]^T$ are reduced to a set of independent parameters $\mathbf{F} = [\mathbf{F}_1, \mathbf{F}_2]^T$. The parameters $\mathbf{F}$ are subsequently optimised. The Fourier coefficients $\hat{\mathbf{\Phi}}$ of $\mathbf{\Phi}$ are reconstructed as $\hat{\mathbf{\Phi}}_{(k_1,k_2)} = \mathbf{A}_{(k_1,k_2)} F_{(k_1,k_2)}$. For modes with $k_1 \neq 0$, $\hat{\mathbf{\Phi}}_1$ is eliminated, $\mathbf{F}_1 = \mathbf{\Phi}_2$, and $\mathbf{F}_2 = \mathbf{\Phi}_3$:

$$\begin{bmatrix} \hat{\mathbf{\Phi}}_{1,(k_1,k_2)} \\ \hat{\mathbf{\Phi}}_{2,(k_1,k_2)} \\ \hat{\mathbf{\Phi}}_{3,(k_1,k_2)} \end{bmatrix} = \left[\begin{array}{c|c} -\frac{k_2}{k_1}\mathbf{I} & \frac{\imath}{k_1}\mathbf{D_z} \\ \mathbf{I} & \\ & \mathbf{I} \end{array} \right] \left[\begin{array}{c} \mathbf{F}_{1,(k_1,k_2)} \\ \hline \mathbf{F}_{2,(k_1,k_2)} \end{array} \right]. \qquad (3.26)$$

For modes with $k_1 = 0$, a similar system is formulated, eliminating $\hat{\mathbf{\Phi}}_2$:

$$\begin{bmatrix} \hat{\mathbf{\Phi}}_{1,(0,k_2)} \\ \hat{\mathbf{\Phi}}_{2,(0,k_2)} \\ \hat{\mathbf{\Phi}}_{3,(0,k_2)} \end{bmatrix} = \left[\begin{array}{c|c} \mathbf{I} & \\ & \frac{\imath}{k_2}\mathbf{D_z} \\ & \mathbf{I} \end{array} \right] \left[\begin{array}{c} \mathbf{F}_{1,(0,k_2)} \\ \hline \mathbf{F}_{2,(0,k_2)} \end{array} \right]. \qquad (3.27)$$

Energy constraint

The second constraint that needs to be imposed on the parameters $\mathbf{\Phi}$ dictates that the energy per unit volume of the perturbations should remain on a predefined level $E_0(1/2)$. This is a non-linear constraint; expressed on the discretised parameter set $\mathbf{F}$, and using the Parseval identity (Canuto et al., 2006), it corresponds to

$$
\begin{aligned}
& C(\mathbf{F}) \qquad (3.28) \\
\equiv \quad & \frac{E_0}{2} - \frac{1}{2L_3} \sum_{\frac{k_1 L_1}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \sum_{\frac{k_2 L_2}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \hat{\mathbf{\Phi}}^H_{(k_1,k_2)} \mathbf{\Omega}\, \hat{\mathbf{\Phi}}_{(k_1,k_2)} = 0 \\
= \quad & \frac{E_0}{2} - \frac{1}{2L_3} \sum_{\frac{k_1 L_1}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \sum_{\frac{k_2 L_2}{2\pi}=-\frac{N}{2}}^{(\frac{N}{2}-1)} \mathbf{F}^H_{(k_1,k_2)} \mathbf{A}^H_{(k_1,k_2)} \mathbf{\Omega}\, \mathbf{A}_{(k_1,k_2)} \mathbf{F}_{(k_1,k_2)} = 0,
\end{aligned}
$$

with $\mathbf{\Omega}$ a diagonal matrix containing the fourth order interpolated cell volumes in the normal direction, and superscript H denoting the Hermitian transpose.

The energy constraint is imposed with one of the following methods: the augmented Lagrangian method or the gradient projection method. The methods, explained in section 2.5, are applied to the reduced parameter set $\mathbf{F}$ instead of $\mathbf{\Phi}$. The results obtained with these methods are compared in section 4.2. For the optimisation of the turbulent mixing layer, we show that the gradient-projection method yields a more robust optimisation algorithm.

3.3.2 Regularisation

Regularisation is required, since to remain relevant from a physical point of view the perturbation and their updates have to be sufficiently smooth. From the mathematical point of view it is unknown what properties the initial field, and thus the perturbations, should fulfil to lead to a well-posed Navier–Stokes problem (Protas et al., 2004). Protas et al. (2004) assume that the fields belong to the $C^\infty(0,T;C^\infty(\Omega))$, which means that they are infinitely differentiable in time and space. In case of direct numerical simulations the field is approximated on a grid that is fine enough to allow to represent the velocity field up to the smallest scales that have a significant amount of energy. This restricts the perturbations to frequencies a factor lower than the Nyquist frequency, or the perturbations should be smooth in space.

In the context of shape optimisation Jameson et al. (1998) desired the optimal shape to be smooth. To this end they introduced smoothing based on a second-order differential filter, given by a Helmholtz equation i.e.

$$\overline{\mathcal{G}} - \epsilon \frac{\partial^2 \overline{\mathcal{G}}}{\partial x_3 \partial x_3} = \mathcal{G}, \tag{3.29}$$

with $\overline{\mathcal{G}}$ the smoothed gradient and ϵ a smoothing parameter. Jameson and Vassberg (2001) observed that introducing the filter fastened up the convergence of the optimisation.

For a 1 D problem with periodic boundary conditions, the solution of equation (3.29), the smoothed gradient, corresponds to:

$$\hat{\overline{\mathcal{G}}} \quad = \quad \frac{1}{(1+\epsilon k^2)} \hat{\mathcal{G}} = \frac{k_c^2}{(k_c^2 + k^2)} \hat{\mathcal{G}} \tag{3.30}$$

as $k_c^2/(k_c^2+k^2)$ corresponds to a low-pass filter with cut-off wavenumber $k_c = 1/\sqrt{\epsilon}$, so the smoothed gradient is the low-pass filtered gradient.

For the current work solving the equation (3.29) requires calculating the inverse of $\mathbf{I} - \alpha \mathbf{D_z^T D_z}$ (with $\mathbf{D_z}$ the matrix representing the finite-difference discretisation of $\partial/\partial x_3$ in the orthogonal direction). To avoid this, we filter the three directions separately with 1D low-pass filters. In the spanwise and streamwise directions our choice of discretisation projects the solution onto a Fourier basis. In these directions a low-pass sharp spectral cut-off filter is applied by selecting for the optimisation only the n modes with longest wavelength. In the normal direction, where a finite-difference discretisation is used, we use a Gaussian filter:

$$\overline{\mathcal{G}}(x_3) = \left(\frac{6}{\pi\Delta^2}\right)^{1/2} \int_{-L_3/2}^{L_3/2} exp\left(-\frac{6(x_3 - x')^2}{\Delta^2}\right) .\mathcal{G}(x')\, \mathrm{d}x' \tag{3.31}$$

with Δ the filter width taken equal to 10 cells. The variance of the filter is $\sigma = \Delta/\sqrt(12)$ such that the Gaussian filter has the same second moments ($\int_{-\infty}^{\infty} x^2 G(x)\, \mathrm{d}x$ for filter function $G(x)$, i.e. $G(x) = \sqrt{(6/\pi\Delta^2)}exp(-6x^2/\Delta^2)$) as a box filter with the filter width Δ (Pope, 2000). The filter is applied on the gradient in the reduced parameter space, the space of $\mathbf{F}$. The Gaussian filter is compared with the filter of (3.30) in figure 3.5. We found that the Gaussian filter sufficiently regularises the gradient, and further increased the rate of convergence of the optimisation significantly.

Protas et al. (2004) and Heinkenschloss[1] show that the gradient can also be smoothed by selecting an inner product different from the L_2 inner product to define a smoothed gradient based on the adjoint method. The smoothing method

---

[1] private communication at the occasion the visit of professor Heinkenschloss to the K.U.Leuven Optimization in Engineering Center (OPTEC) as speaker for the Simon Stevin Lecture in March 2010

will be outlined here, and the method will be linked to the low pass filter. For simplicity, the derivation uses the gradient in the parameter space of $\boldsymbol{\phi}$, and not gradient in the reduced parameter space, the space of $\mathbf{F}$.

With the adjoint method the non-smoothed gradient, $\mathcal{G}$, is defined by the reformulated Gateaux differential (3.5) using an L_2 inner product (2.34), as described on p. 32:

$$\delta \mathscr{J}(\boldsymbol{\phi}; \delta\boldsymbol{\phi}) = (\boldsymbol{u}^*, \delta\boldsymbol{\phi})_0 = (\mathcal{G}, \delta\boldsymbol{\phi})_0 \tag{3.32}$$

Protas et al. (2004) propose to precondition the gradient by using a linear combination of the L_2 and the Hilbert space H^1 inner product instead of only the L_2 inner product on the right hand side of (3.32). They denote the proposed inner product as $W^{l1,\infty}$ (Protas et al., 2004):

$$(a,b)_{W^{l1,\infty},t_i} = \int_\Omega \frac{1}{(1+l_1^2)} \left(\boldsymbol{a}(\mathbf{x},t_i).\boldsymbol{b}(\mathbf{x},t_i) + l_1^2 \frac{\partial}{\partial x_j}\boldsymbol{a}(\mathbf{x},t_i).\frac{\partial}{\partial x_j}\boldsymbol{b}(\mathbf{x},t_i) \right) \,\mathrm{d}\mathbf{x} \tag{3.33}$$

The smoothed gradient $\overline{\mathcal{G}}$ is defined as:

$$\delta \mathscr{J}(\boldsymbol{\phi}; \delta\boldsymbol{\phi}) = \left(\overline{\mathcal{G}}, \delta\boldsymbol{\phi}\right)_{W^{l1,\infty},0} \tag{3.34}$$

Combination of (3.32) and (3.34) leads to the relationship between $\mathcal{G}$ and $\overline{\mathcal{G}}$:

$$\begin{aligned}
(\mathcal{G}, \delta\boldsymbol{\phi})_0 &= \left(\overline{\mathcal{G}}, \delta\boldsymbol{\phi}\right)_{W^{l1,\infty},0} \\
&= \int_\Omega \frac{1}{(1+l_1^2)} \left(\overline{\mathcal{G}}(\mathbf{x},t_i).\delta\boldsymbol{\phi}(\mathbf{x},t_i) - l_1^2 \frac{\partial^2}{\partial x_j^2}\overline{\mathcal{G}}(\mathbf{x},t_i).\delta\boldsymbol{\phi}(\mathbf{x},t_i) \right) \,\mathrm{d}\mathbf{x} \\
&= \left(\frac{1}{(1+l_1^2)} \left(\overline{\mathcal{G}} - l_1^2 \frac{\partial^2}{\partial x_j^2}\overline{\mathcal{G}} \right), \delta\boldsymbol{\phi} \right)_0
\end{aligned} \tag{3.35}$$

The smoothed gradient is then identified by solving the Helmholtz equation (Protas et al., 2004):

$$\frac{1}{(1+l_1^2)} \left(\overline{\mathcal{G}} - l_1^2 \frac{\partial^2}{\partial x_j^2}\overline{\mathcal{G}} \right) = \mathcal{G} \tag{3.36}$$

This equation is similar to Eq.(3.29), so low pass filtering can be obtained by using the $W^{l1,\infty}$ inner product instead of the L_2 inner product.

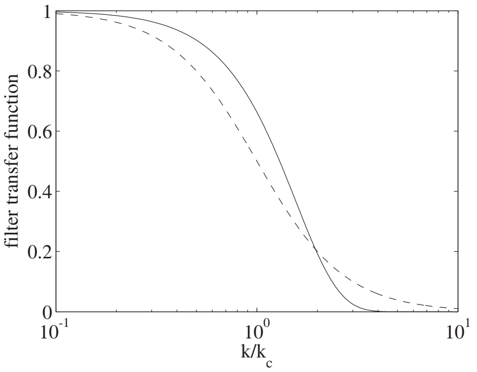


Figure 3.5: Transfer functions of the Gaussian filter (–) and the Helmholtz filter (−−) with k_c the cut-off frequency

Chapter 4

Verification of the optimisation method

The numerical performance of the optimisation method is evaluated. First the adjoint calculation of the gradient is verified with a convergence study, secondly the two methods to handle the energy constraint are compared. Subsequently optimisation is performed to five different cost functional formulations. The work discussed in this chapter is published in Delport, Baelmans, and Meyers (2008); Delport et al. (2009).

4.1 Verification of the adjoint-based gradient

As discussed in section 2.4 and 3.2.2, the use of a continuous adjoint methodology may introduce an inconsistency between the gradient obtained from the adjoint equations and the gradient of the discretised cost functional. This may be further increased by the smoothing of the gradient in the orthogonal direction (cf. section 3.3.2). In the current section, we will first evaluate the difference between the forward and adjoint calculated gradient without the use of smoothing. Proper convergence of the adjoint-based gradient for decreasing time steps and mesh spacings is verified. Subsequently, the influence of the gradient smoothing is discussed.

The forward evaluation uses the central differencing method. This requires Navier–Stokes calculations per parameter direction, i.e., $\boldsymbol{u}(\boldsymbol{\phi} + \alpha\delta\boldsymbol{\phi})$ and $\boldsymbol{u}(\boldsymbol{\phi} - \alpha\delta\boldsymbol{\phi})$ with α small. In order to limit the computational time of the forward gradient calculation (proportional to the number of parameter directions $\delta\boldsymbol{\phi}$), we have limited the verification of the gradient to a few parameter directions only. The forward evaluation results in the value of the derivative for all time horizons up to the end of the simulation time. This is not the case for the adjoint-based gradient, as the latter results from a Navier-Stokes calculation up to the time horizon followed by an adjoint calculation back in time. The adjoint gradient is only calculated for time horizons $T = 0, 5, \dots, 30$. The two derivatives are compared and the difference between the two is analysed as a function of the grid and the time step. Before starting the analysis, the computational set-up is outlined.

For the point $\boldsymbol{\phi}$ in which the gradients are calculated, we selected a composition of two-dimensional and three-dimensional perturbations. The two-dimensional perturbations are a superposition of the most unstable eigenfunction according to the linear stability theory (LST) for respectively Fourier index $(4, 0)$ and $(2, 0)$ (here the Fourier index (α, β) corresponds to wavenumber $(\alpha 2\pi/L_1, \beta 2\pi/L_2)$ with $L_1 = 4\lambda_1$, and $L_2 = 4\lambda_2$ (cf. section 3.2.1)). They are combined with three-dimensional streamwise invariant perturbations with Fourier index $(0, 4)$. The profile in the normal direction of the $(0, 4)$ mode corresponds with a Gaussian distribution on the vorticity according to Moser and Rogers (1993). Results are shown here for the evaluation of the gradient in a direction $\delta\boldsymbol{\phi}$ which is parallel to $\boldsymbol{\phi}$, such that the derivative towards the amplitude of the parameters is calculated. Results obtained for other directions (not shown here), show similar trends. Several cost functionals are considered, i.e. based on momentum thickness, turbulent kinetic energy, mean-flow kinetic energy, total kinetic energy, and enstrophy. For a discussion of their physical relevance, etc., we refer to section 4.3.

Figure 4.1 compares the derivative of a cost functional $\mathscr{J}(T)$ based on the turbulent kinetic energy (TKE) as function of the simulation time window T. The figure depicts the derivative calculated using a forward finite-difference method and an adjoint-based method. Simulations are performed on a $32 \times 20 \times 1024$ uniform grid, with a CFL number of 0.1. It is observed that the adjoint-based calculation corresponds well to the gradient of the discrete forward system for this selection of grid and CFL number.

In figure 4.2, the error between the gradient of the forward system and the adjoint-based calculation is presented as function of CFL number and grid resolution. Results for five different cost functionals are included. The error is calculated as a normalised L_2 norm of the difference between the forward and adjoint gradient, evaluated at seven different time horizons $T = 0, 5, \dots, 30$:

$$\varepsilon = \frac{||\delta \mathscr{J}_{adj} - \delta \mathscr{J}_{FD}||_2}{||\delta \mathscr{J}_{FD}||_2}. \tag{4.1}$$

First, in figure 4.2 (a), the effect of the CFL number is considered. The grid is kept constant at $32 \times 20 \times 1024$. In the absence of smoothing, a clear first-

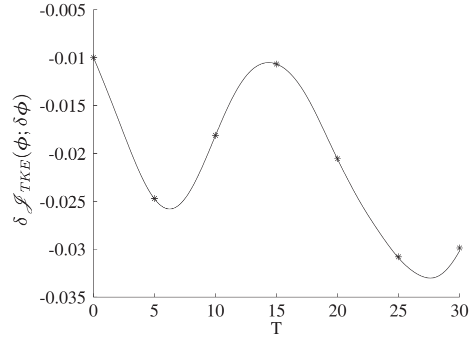
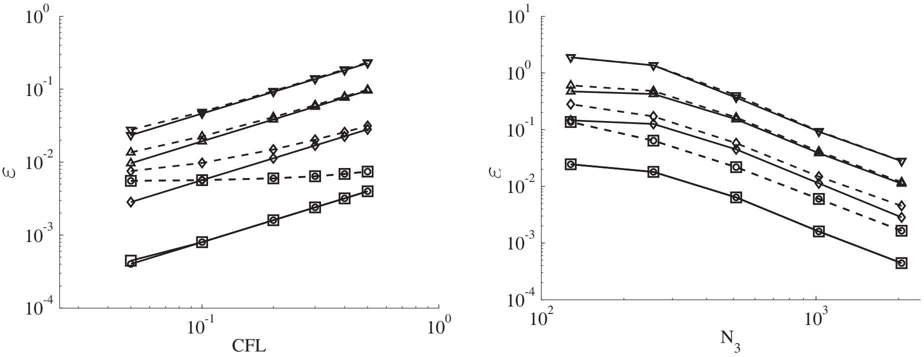


Figure 4.1: Derivative of cost functional based on turbulent kinetic energy (cf. Eq. 4.11) towards the amplitude of a selected set of perturbations (CFL=0.1, N_3=1024), forward finite-difference evaluation (—), adjoint-based evaluation ($*$).

order convergence of the difference is observed. This is in line with the piecewise constant approximation of the non linear forward field in the Runge–Kutta time integration of the adjoint system (cf. discussion in section 3.2.2). When smoothing is added to the gradient, the difference caused by the filtering outweighs the difference due to the time integration for the lower CFL numbers.

In figure 4.2 (b), the effect of the grid in the normal direction is considered, while the CFL number is kept constant at 0.2. The difference decreases with order 1.7. This corresponds indirectly to the first-order time integration error. We found that the time step (for finer normal grids N_3) is limited by the diffusive CFL number (3.18), and hence $\Delta t \sim \mathrm{CFL}_{\mathit{diff}}/N_3^2$. As a consequence the time step is related to the mesh spacing squared. Since the time integration is first order in the time step, an order close to two is observed for the grid refinement. When the smoothed gradient is considered in figure 4.2 (b), no change in order is observed, since this smoothing is based on a second-order filter operator (cf. section 3.3.2)

This test showed first order convergence of the adjoint gradient in function of the time step and found convergence with order 1.7 as a function of the grid spacing in the x_3-direction. The error on the adjoint gradient is small, though its relative magnitude is non negligible; this will impact the performance of the augmented Lagrangian method, as discussed in the next section.



(a) CFL (b) grid

Figure 4.2: The normalised L_2 norm ε of the differences between the adjoint-based and the forward finite-difference gradient (FD) evaluated for simulation time horizon $T = 0, 5, \ldots, 30$ as function of the CFL-number and the normal grid. Adjoint-based gradients without regularisation (—); with regularisation (−−). Cost functionals are based on: momentum thickness (○); turbulent kinetic energy (◊); mean-flow kinetic energy (□); total kinetic energy (△); and enstrophy (▽).

4.2 Comparison of the augmented Lagrangian and gradient projection method

We now focus on the verification and comparison of the optimisation with the two methods to impose the energy constraint: the augmented Lagrangian and the gradient projection method. The performance of the optimisation with either of the two constraint handling methods is compared. For this test a cost functional is used that maximises the turbulent kinetic energy at the time horizon T (formulated as a minimisation problem):

$$\mathscr{J}_{TKE} = -\frac{1}{2}\frac{1}{\Omega}\left(\boldsymbol{u} - \langle \boldsymbol{u} \rangle, \boldsymbol{u} - \langle \boldsymbol{u} \rangle\right)_T . \tag{4.2}$$

Simulations are performed on a $64 \times 64 \times 128$ uniform grid, and a CFL number of 0.5 is employed. The starting point $\mathbf{F}^{(0)}$ for the optimisation is a combination of 40 Fourier-modes, i.e. with index (α, β), $(0, \beta)$, and $(\alpha, 0)$ and $\alpha, \beta = \mp 1, 2, 3, 4$ (the Fourier index (α, β) corresponds to wavenumber $(\alpha 2\pi/L_1, \beta 2\pi/L_2)$; the profiles of (α, β) and $(\alpha, -\beta)$ are complex conjugates, see section 3.2.2). For each mode, Gaussian profiles with width $\sigma = 3.5$ are used in the normal direction, i.e.:

$$\mathbf{F}_{1,(k_1,k_2)}(x_3) = \mathbf{F}_{2,(k_1,k_2)}(x_3) = A_{(k_1,k_2)} \cdot e^{-x_3^2/(2\sigma^2)} cos(k_1 x_1 + k_2 x_2) \tag{4.3}$$

The amplitude A is determined such that the total initial energy is evenly distributed over the 40 modes. The imposed energy level per unit volume is $E_0(1/2)$ with $E_0 = 10^{-3}$ and $1/2$ equal to the mean-field energy per unit volume of a box that is infinite in the normal direction. The time horizon is set to 20.

In figure 4.3 the convergence history of the optimisation with the gradient projection method and the augmented Lagrangian method is shown. Figure 4.3 (a) displays the evolution of the cost functional, and figure 4.3 (b) plots the constraint violation. First of all, it is appreciated from this figure that optimisation using gradient projection for the constraint, leads to a fast decrease in cost functional during the first 30 iterations, afterwards the improvement slows down. At iteration number 200, we stopped the optimisation. At this point, the relative improvement obtained by the line search is 10^{-5}. An advantage of the gradient projection method is that the optimisation cycle can be stopped at any time, since the constraint is strictly enforced.

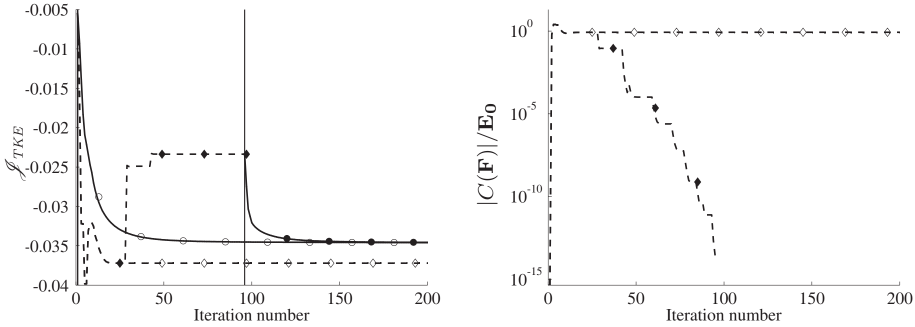
In figure 4.3 (a) results using the standard textbook implementation of the augmented Lagrangian method (cf. Nocedal and Wright, 2006) are also presented. We found that no convergence could be obtained, and the energy constraint in particular, remained violated. The cost functional decreased to a level which is lower than that obtained using the gradient-projection method, but this is an effect of the violation of the energy constraint (fig. 4.3 (b)). This violation leads to initial perturbations containing too much energy, resulting in more turbulent kinetic energy at the time horizon, and a lower value of the cost functional.

This convergence problem is related to the accuracy of the adjoint-based gradient and the stop criterion of the sub-problem $\mathscr{L}_{\lambda^{(n)},\mu^{(n)}}$ (cf. section 2.5). The stop criterion for the optimisation of the sub-problem is:

$$|\nabla \mathscr{L}_{\lambda^{(n)},\mu^{(n)}}(\mathbf{F})| = |\nabla \mathscr{J}(\mathbf{F}) + \Big(-\lambda^{(n)} + \mu^{(n)} C(\mathbf{F})\Big) \nabla C(\mathbf{F})| < \omega^{(n)}. \quad (4.4)$$

This stop criterion relies on the magnitude of the gradient. We opted to calculate the gradient with the adjoint method. In section 4.1, a comparison between the results of the adjoint and the direct gradient calculation (gradient of the discretised cost functional) was presented. Though the difference was found to be small, its relative magnitude is non-negligible. Consequently, a stop criterion which is solely based on the gradient of the cost functional is not always robust: e.g., due to small gradient inconsistencies, the adjoint-based gradient may point in a direction in which a subsequent line-search based on the full non-linear equations may not find a new optimum. The convergence problem observed in figure 4.3 (a) results from a situation where $|\nabla \mathscr{L}| > \omega^{(n)}$, but where the line-search algorithm (cfr. section 2.3.2), which solves the forward equations for function evaluations, finds no further improvement in the direction of the adjoint-based gradient.

Obviously, the stop criterion for $\mathscr{L}$ is easily tweaked to circumvent above-discussed



(a) Cost functional

(b) Relative error on the constraint

Figure 4.3: Comparison of the gradient projection (—○), standard augmented Lagrangian method (– – ◇), adapted augmented Lagrangian method (– – ♦), gradient projection started from optimum attained by the adapted augmented Lagrangian method (—●). (a): evolution of the cost functional; (b) evolution of the energy constraint (for the augmented Lagrangian).

convergence issue. To this end, we stop the sub-problem $\mathscr{L}_{\lambda^{(n)},\mu^{(n)}}$ when a direct evaluation of the sensitivity of the cost functional (using the line-search algorithm) in the search direction is smaller than $\omega^{(n)}/2$:

$$\frac{|\mathscr{J}\left(\mathbf{F^{(k)}}\right) - \mathscr{J}\left(\mathbf{F^{(k-1)}}\right)|}{|\mathbf{F^{(k)}} - \mathbf{F^{(k-1)}}|} < \frac{\omega^{(n)}}{2} \tag{4.5}$$

with (k) the conjugate-gradient iteration number. Results using this adapted stop criterion are also displayed in figure 4.3. Now, the augmented Lagrangian converges after only 96 iterations, and the energy constraint is satisfied up to 10^{-14}. However, the value of the cost functional is now higher than the one obtained using the gradient projection algorithm. Moreover, when the converged solution of the augmented Lagrangian is used as a starting point for the gradient-projection method (as also shown in figure 4.3 (a)), we observe that the cost functional converges further to the same level as the pure gradient-projection optimisation, so the adapted augmented Lagrangian failed to converge to a local optimum.

We conclude that the use of a gradient-projection methodology, to enforce the energy constraint, provides a more robust optimisation platform in the context of a continuous adjoint methodology. The gradient-projection methodology is less sensitive to discretisation errors affecting the gradient calculation. For this reason we use the gradient projection method in the remaining part of this work.

4.3 Optimisation with different cost functionals

Five different cost functionals are defined based on properties that are often monitored in the temporal mixing layer: the momentum thickness, turbulent kinetic energy, mean-flow kinetic energy, total kinetic energy, and enstrophy. As a test for the selected optimisation method, and to test the impact of the cost functional, optimisation is performed towards these cost functionals in a fully three-dimensional turbulent mixing layer with up to $\mathcal{O}(10^4)$ free parameters and a linear and non-linear constraint on the parameters. The results are discussed over two different time horizons and the optima for the different cost functionals are compared. It is found that the momentum thickness, turbulent kinetic energy and mean-flow kinetic energy cost functionals lead to mainly two-dimensional vortex structures, while the total kinetic energy and enstrophy cost functional promote small-scale structures in the flow. The cost functionals leading to small-scale structures are used in chapter 5 and 6 for optimisation to longer time horizons.

4.3.1 Cost functionals

The first candidate cost functional is the momentum thickness. It measures the width of the interaction zone between the upper and lower fluid streams, with respectively free-stream velocity $U_1 = \Delta U/2$ and $U_2 = -\Delta U/2$ (Nygaard and Glezer, 1991; Rogers and Moser, 1994; Balaras et al., 2001):

$$\theta = \frac{1}{(\Delta U)^2} \int_{-L_3/2}^{L_3/2} (\langle u_1(\mathbf{x},T)\rangle - U_2)(U_1 - \langle u_1(\mathbf{x},T)\rangle)\,\mathrm{d}x_3 \tag{4.6}$$

$$= \int_{-L_3/2}^{L_3/2} \left(\frac{1}{4} - \left(\frac{\langle u_1(\mathbf{x},T)\rangle}{\Delta U}\right)^2\right)\mathrm{d}x_3, \tag{4.7}$$

with $\langle u_1(\mathbf{x},T)\rangle$ the streamwise velocity averaged over the plane of homogeneity of the flow $(x_1 - x_2)$. The momentum thickness grows linearly in time in the self-similar region (Rogers and Moser, 1994; Balaras et al., 2001; Pope, 2000). Consequently, the cost functional maximises the momentum thickness. Formulated as a minimisation problem, as is the convention in optimisation, this yields

$$\mathscr{J}_{M} = -\frac{1}{L_1 L_2} \int_{\Omega} \left(\frac{1}{4} - \left(\frac{\langle u_1(\mathbf{x},T)\rangle}{\Delta U}\right)^2\right)\mathrm{d}\mathbf{x}. \tag{4.8}$$

A set of other cost functionals is based on energy properties: the total kinetic energy, the mean-flow kinetic energy, and the turbulent kinetic energy. The first two properties decrease linearly in time in the self-similar region, while the turbulent kinetic energy increases linearly at the same moment (Rogers and Moser, 1994; Balaras et al., 2001; Pope, 2000). We formulate three cost functionals, which respectively minimise total kinetic energy, minimise mean-flow kinetic energy, and maximise turbulent kinetic energy.

The cost functional minimising total kinetic energy corresponds to (using the inner product notation introduced in (2.34)):

$$\mathscr{J}_{KE} = \frac{1}{2}\frac{1}{\Omega}\left(\boldsymbol{u},\boldsymbol{u}\right)_T \tag{4.9}$$

Similarly, the cost functional minimising mean-flow kinetic energy is

$$\mathscr{J}_{MFE} = \frac{1}{2}\frac{1}{\Omega}\left(\langle\boldsymbol{u}\rangle,\langle\boldsymbol{u}\rangle\right)_T . \tag{4.10}$$

Finally, the cost functional maximising turbulent kinetic energy (formulated as a minimisation problem) is

$$\mathscr{J}_{TKE} = -\frac{1}{2}\frac{1}{\Omega}\left(\boldsymbol{u}-\langle\boldsymbol{u}\rangle,\boldsymbol{u}-\langle\boldsymbol{u}\rangle\right)_T . \tag{4.11}$$

An alternative cost functional, which measures the small-scale gradients in the flow, can be based on the volume averaged enstrophy $\mathscr{E} = \left(\boldsymbol{\omega},\boldsymbol{\omega}\right)_T/(2\Omega)$, with $\boldsymbol{\omega}$ the vorticity ($\omega = \nabla\times\boldsymbol{u}$). We maximise the enstrophy with the cost functional:

$$\mathscr{J}_{ENS} = -\frac{1}{2}\frac{1}{\Omega}\left(\boldsymbol{\omega},\boldsymbol{\omega}\right)_T . \tag{4.12}$$

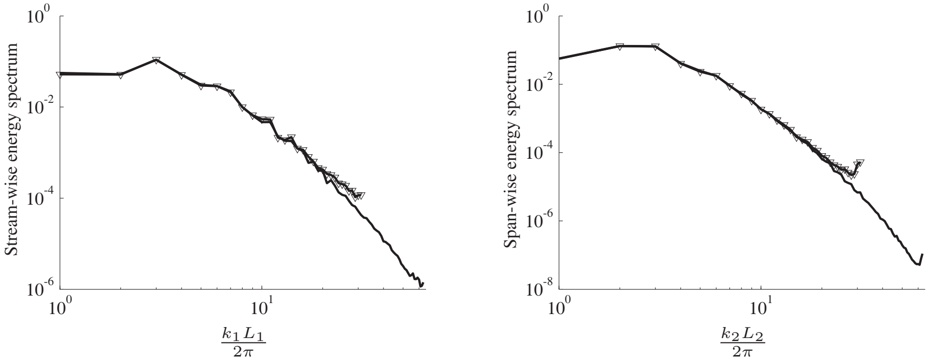
Finally, as discussed in section 2.4.1 the cost functional determines the adjoint initial field at time T, see (2.41). These fields, and their respective cost functionals are given in table 4.1.

4.3.2 Optimisation with different cost functionals and different time horizons

We will now evaluate the cost functionals, presented in the previous section, based on optimisation results. Two time horizons $T = 20$ and $T = 40$ are used. All simulations are performed on a $64\times64\times128$ uniform mesh, with a CFL number of 0.5. In order to verify grid convergence, we compared the flow evolution on the $64\times 64\times128$ mesh and a $128\times128\times256$ mesh when the perturbations optimised to the enstrophy cost functional $\mathscr{J}_{ENS}$ are used. In figure 4.4, streamwise and spanwise energy spectra are presented for both resolutions, showing that the $64\times64\times 128$ solution is well resolved. Compared to other cost functionals, the enstrophy optimal parameters lead to the smallest scales during the flow evolution (see also

Name	$\mathscr{J}(\mathbf{u})$	$\boldsymbol{u}^*(\mathbf{x},T)$
KE	$\frac{1}{2}\frac{1}{\Omega}\left(\boldsymbol{u},\boldsymbol{u}\right)_T$	$\frac{1}{\Omega}\boldsymbol{u}(\mathbf{x},T)$
TKE	$-\frac{1}{2}\frac{1}{\Omega}\left(\boldsymbol{u}-\langle\boldsymbol{u}\rangle,\boldsymbol{u}-\langle\boldsymbol{u}\rangle\right)_T$	$-\frac{1}{\Omega}(\boldsymbol{u}(\mathbf{x},T)-\langle\boldsymbol{u}(\mathbf{x},T)\rangle)$
MFE	$\frac{1}{2}\frac{1}{\Omega}\left(\langle\boldsymbol{u}\rangle,\langle\boldsymbol{u}\rangle\right)_T$	$\frac{1}{\Omega}\langle\boldsymbol{u}(\mathbf{x},T)\rangle$
ENS	$-\frac{1}{2}\frac{1}{\Omega}\left(\boldsymbol{\omega},\boldsymbol{\omega}\right)_T$	$\frac{1}{\Omega}\nabla^2\boldsymbol{u}(\mathbf{x},T)$
M	$-\frac{1}{L_1L_2}\int_\Omega\left(\frac{1}{4}-\left(\frac{\langle u_1(\mathbf{x},T)\rangle}{\Delta U}\right)^2\right)\,\mathrm{d}\mathbf{x}$	$\left[\frac{2}{L_1L_2}\frac{\langle u_1(\mathbf{x},T)\rangle}{(\Delta U)^2},0,0\right]$

Table 4.1: Overview of cost functionals and corresponding initial conditions for the adjoint equations



(a) Streamwise energy spectra

(b) Spanwise energy spectra

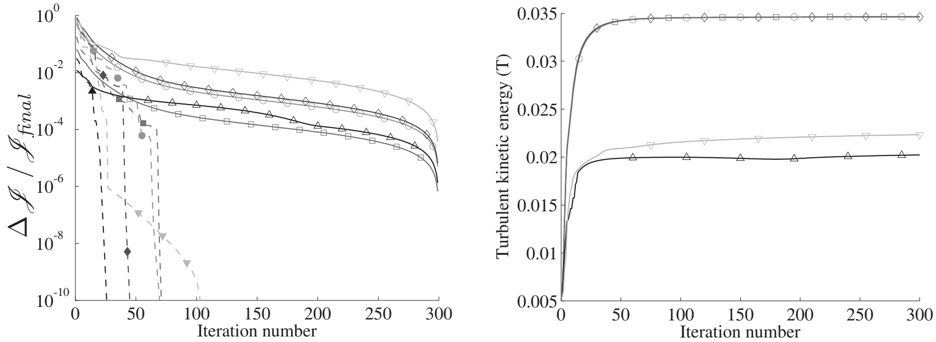
Figure 4.4: Spectra on location $x_3 = 0$ of the flow optimised towards enstrophy for $T = 40$ on a $64 \times 64 \times 128$ mesh ($\triangledown$) and a $128 \times 128 \times 256$ mesh (no symbol). $t = 40$ (a): streamwise energy spectrum, (b): spanwise energy spectrum.

discussion below), such that these convergence results are also representative for the other cases.

First, the convergence history as function of conjugate gradient iterations is presented in figure 4.5 (a). The convergence criterion is based on the relative improvement of the cost functional and set to 10^{-10}. The number of iterations is limited to 300, as each conjugate gradient iteration requires one adjoint simulation for the gradient calculation, and about 8 Navier–Stokes simulations for the line search algorithm. The optimisations to $T = 20$ stopped when they reached the maximum number of iterations. The relative improvement of the cost functional at iteration 300 was 10^{-5}. The optimisations with $T = 40$ converged in less than about 100 iterations.

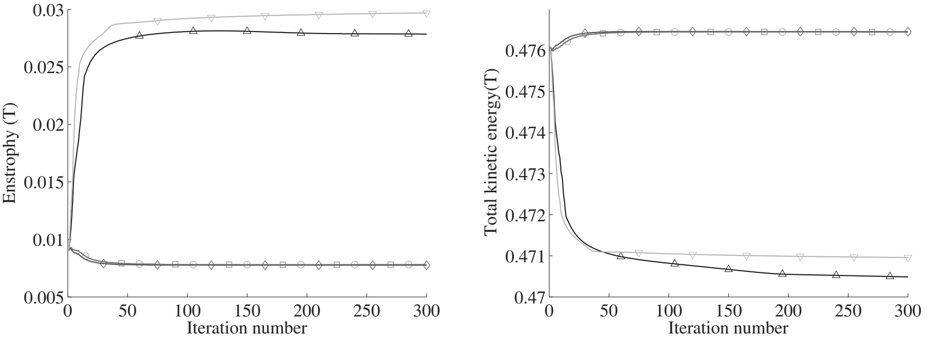
The evolution as function of iteration number of the turbulent kinetic energy, enstrophy, and total kinetic energy evaluated at time $t = 20$, is displayed for $T = 20$ in respectively fig. 4.5 (b), (c), and (d). The figure shows that the optimisation converges along the same path when a cost functional based on momentum thickness $\mathscr{J}_M$, turbulent kinetic energy $\mathscr{J}_{TKE}$, or on mean-flow kinetic energy $\mathscr{J}_{MFE}$, is used. A more precise identification of their respective optima reveals a relative difference (based on a L_2-norm of the difference between their respective optimal perturbations) of 10^{-3} between optima based on momentum thickness versus mean-flow kinetic energy . The relative difference with the optimum based on turbulent kinetic energy is of the order of 0.07. Hence, these three cost functionals lead to similar optima. As further observed in figure 4.5, the optimisations using cost functionals based on total kinetic energy $\mathscr{J}_{KE}$, and enstrophy $\mathscr{J}_{ENS}$ follow different trends. The results for $T = 40$ show similar behaviour (not plotted here).

It is further observed in figure 4.5 for optima obtained using $\mathscr{J}_M$, $\mathscr{J}_{TKE}$, or $\mathscr{J}_{MFE}$, that optimisation improves the level of turbulent kinetic energy at final time $T = 20$, but that enstrophy decreases, and total kinetic energy increases at the time horizon. This indicates that the turbulent kinetic energy for these solutions remains in large-scale modes with low vorticity and low dissipation. In figure 4.6 the flow structures in the solution at $T = 0$ and $T = 20$ are visualised for the different cost functionals using a λ_2 visualisation (Jeong and Hussain, 1995). The λ_2 criterion of Jeong and Hussain (1995) displays the coherent structures or vortices, localisations of rotational motion, in the flow. The λ_2 method locates coherent structures as regions with two negative eigenvalues of the symmetric tensor $S^2 + \Omega^2$ with S, and Ω, respectively the symmetric, and anti-symmetric part of the velocity gradient tensor. The structures are visualised with iso-surfaces of the largest negative eigenvalue (here level -0.005 is used). The streamwise, spanwise and normal direction are respectively indicated as x_1, x_2 and x_3. The mean-velocity profile is also shown, indicating the direction of the flow in the upper and lower stream. In figure 4.6, it is observed that the cost functionals based on momentum thickness, turbulent, or mean-flow kinetic energy lead to large two-



(a) Cost functional $\mathscr{J}$

(b) Turbulent kinetic energy



(c) Enstrophy

(d) Total kinetic energy

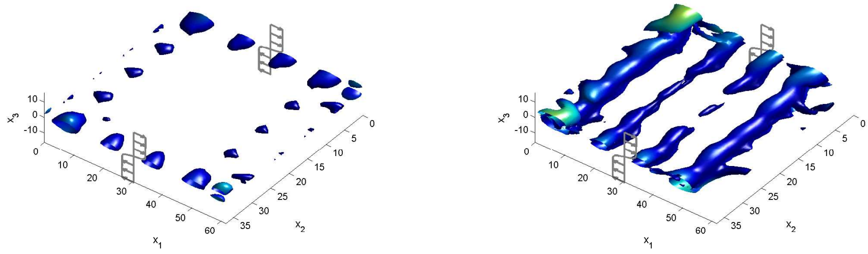
Figure 4.5: Optimisation towards momentum thickness (○), turbulent kinetic energy (◊), mean-flow kinetic energy (□), total kinetic energy (△), and enstrophy (▽). (a) Convergence history (with $\Delta\mathscr{J} = \mathscr{J}(it) - \mathscr{J}_{final}$) for $T = 20$ (—, empty symbols), and $T = 40$ (−−, filled symbols); (b),(c), and (d) Evolution of cost functional properties at T during optimisation procedure for $T = 20$.

dimensional rollers. More careful examination of the two-dimensional rollers in figure 4.6 (b) reveals a slight bending in the spanwise direction, corresponding to a small three-dimensional disturbance on these two-dimensional rollers. However, we found that the cost functionals $\mathscr{J}_{M}$, $\mathscr{J}_{TKE}$, and $\mathscr{J}_{MFE}$ are very insensitive to small variations on the initial perturbation in this 3D spanwise direction, by manually removing the three-dimensional component from the perturbations we find a relative difference on the cost functionals of only 10^{-5}.

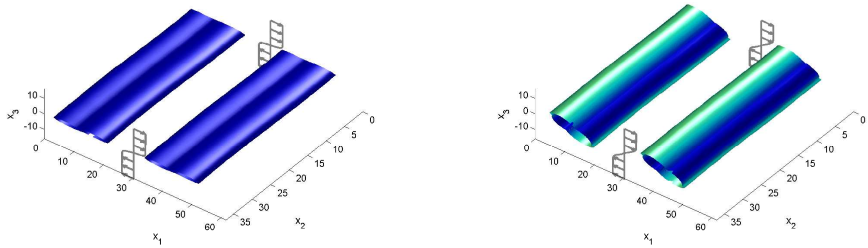
The two-dimensional rollers obtained for $\mathscr{J}_{M}$, $\mathscr{J}_{TKE}$, and $\mathscr{J}_{MFE}$, resemble the typical evolution obtained by superimposing two-dimensional LST modes on the mean flow (e.g., Rogers and Moser, 1992). The energy of the optimised perturbations is concentrated in the four largest streamwise wave numbers, among which the most unstable LST wavenumber and two sub harmonics. However, the profiles of the optimised perturbations do not match in the normal direction with the most unstable LST eigenfunctions at these wave numbers. This is not unexpected for two reasons. First of all, the roll-up of mixing-layer perturbations into large two-dimensional vortices is a non-linear phenomenon, such that a non-linear optimisation may yield results different from linear-stability analysis. Secondly, as, e.g., pointed out in Schmid and Henningson (2001), also in the linear regime, a final-time optimal growth problem is not a mere superposition of the most unstable eigenfunctions, since LST eigenfunctions are non orthogonal. The effect of non orthogonality of the eigenfunctions is schematically illustrated for two modes in figure 4.7. It shows that distributing the energy over several non-orthogonal eigenmodes can lead to an initially faster growing instability than injecting all energy in the most unstable mode due to the angle between the eigenmodes.

To further investigate the effect of the profiles, we simulated the flow using perturbations based on the most unstable LST eigenfunctions at wave numbers $k_1 L_1/(2\pi) = 4, 3, 2,$ and 1, with the same energy distribution for these wave numbers as obtained from the $\mathscr{J}_{TKE}$ optimisation. We found the ensuing turbulent kinetic energy at $t = 20$ to be about 30% lower than the one obtained with the optimised perturbations.

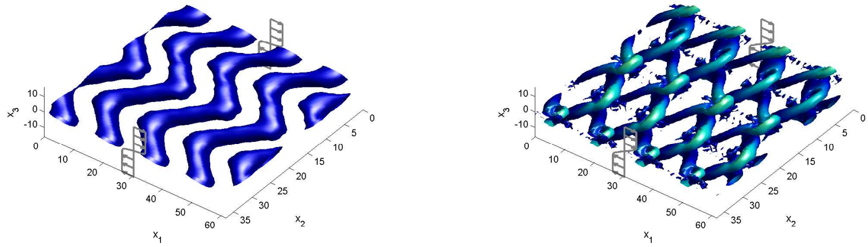
In figure 4.6 and 4.8, the flow structures of the optimised solutions are visualised for the different cost functionals, respectively with time horizon $T = 20$ and $T = 40$. The two-dimensional large-scale vortices which emerge when $\mathscr{J}_{M}$, $\mathscr{J}_{TKE}$, or $\mathscr{J}_{MFE}$, is used, are apparent for both time horizons. $\mathscr{J}_{KE}$ and $\mathscr{J}_{ENS}$ lead to much finer scales in the solution at the final time window. For $\mathscr{J}_{KE}$ and $T = 40$ the fine three-dimensional structures are combined with large two-dimensional rollers. To better assess the broad-banded nature of $\mathscr{J}_{KE}$ and $\mathscr{J}_{ENS}$ optima for $T = 40$, we evaluate the streamwise and spanwise energy spectra of both solutions in figure 5.7 at time $t = 0$ and $t = 40$. While the energy in the optimal initial perturbations ($t = 0$) is distributed over the low wavenumber modes only, the spectra at $t = 40$ are clearly more broad-banded.



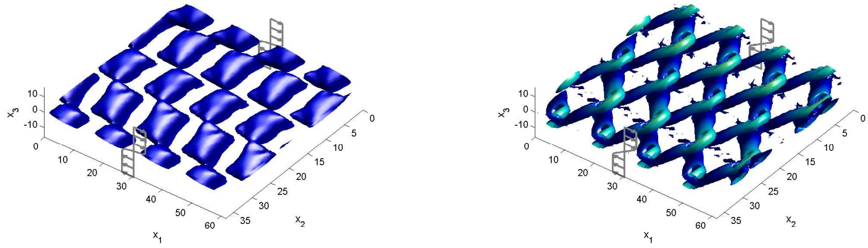
(a) Unoptimised



(b) Turbulent kinetic energy $\mathscr{J}_{TKE}$(momentum thickness $\mathscr{J}_{M}$ and mean-flow kinetic energy $\mathscr{J}_{MFE}$ are similar)

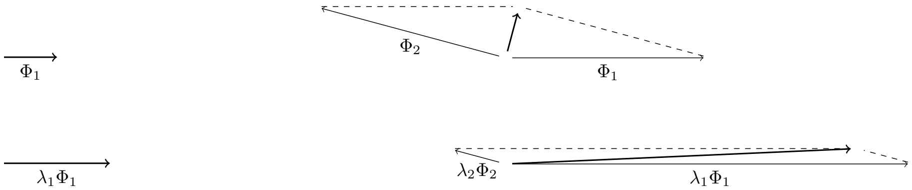


(c) Total kinetic energy $\mathscr{J}_{KE}$



(d) Enstrophy $\mathscr{J}_{ENS}$

Figure 4.6: λ_2 visualisation of the velocity-field coloured with the distance from the mixing centre line for parameters optimised to the different cost functionals with $T = 20$; profile $\langle u_1 \rangle$ (—); visualisation of $t = 0$ (left) and of $t = T$ (right)



(a) all energy in most unstable mode Φ_1

(b) energy distributed over modes Φ_1 and Φ_2

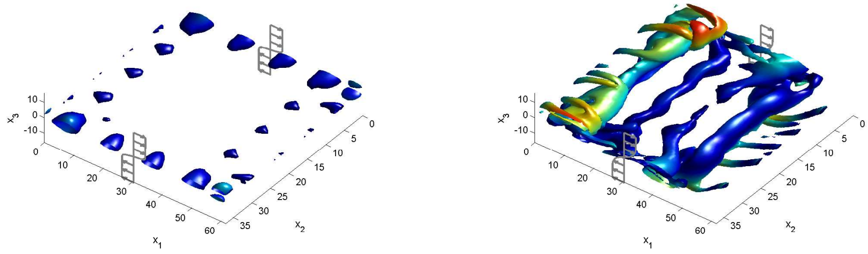
Figure 4.7: Combination of non orthogonal eigenmodes can lead to a faster growing instability than the most unstable mode Φ_1. Initial vectors have the same size and thus energy. The eigenmodes Φ_1 and Φ_2 have respectively eigenvalue λ_1 and λ_2.

In figure 4.10, we evaluate the time history ($t = 0 - 80$) of solutions optimised for a time window $T = 20$ and $T = 40$. Turbulent kinetic energy is represented for optimisation with $\mathscr{J}_{TKE}$ in fig. 4.10 (a), enstrophy for $\mathscr{J}_{ENS}$ in fig. 4.10 (b), and total kinetic energy for $\mathscr{J}_{KE}$ in fig. 4.10(c). For the optimisation to $\mathscr{J}_{TKE}$ and $\mathscr{J}_{ENS}$ it is observed that the optimum for $T = 20$ is not optimal for $T = 40$. The evolution of turbulent kinetic energy in fig. 4.10 (a), and enstrophy in fig. 4.10 (b), both show a peak, which is 'shifted' towards $t = 20$ or $t = 40$ depending on the optimisation time window T. This effect is not found for flow optimised to $\mathscr{J}_{KE}$ (fig. 4.10(c)), and the total kinetic energy decreases monotonically in time.

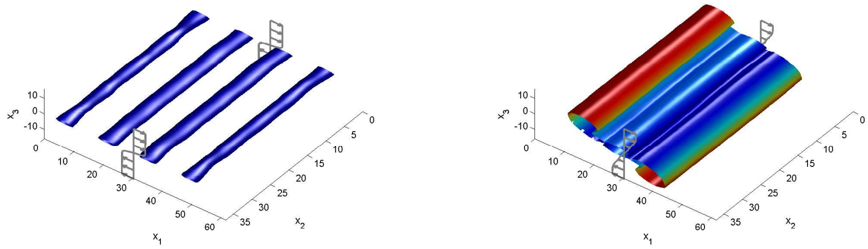
Finally, in figure 4.10(c) we observe that the optimum obtained using $\mathscr{J}_{KE}$ and $T = 20$ evaluated at time $t = 40$ is better than direct optimisation for time horizon $T = 40$ at time $t = 40$. This points to the existence of local minima, a result of the topologically complex parameter space in which we optimise (Bewley et al., 2001). To further establish this, the optimal perturbations obtained for $T = 20$ were used as a starting point for optimisation with $T = 40$, and it was found that the optimisation with time window $T = 40$ now converged to a set of perturbations which are close to those found for $T = 20$. To avoid this, it can be opted to start the optimisation to $T = 40$ systematically from the optima for $T = 20$. This procedure is explained in more detail in the section 5.2. Obviously, this procedure can not exclude local optima. The existence of local optima can only be verified by systematically varying the starting point of the optimisation.

4.4 Conclusion

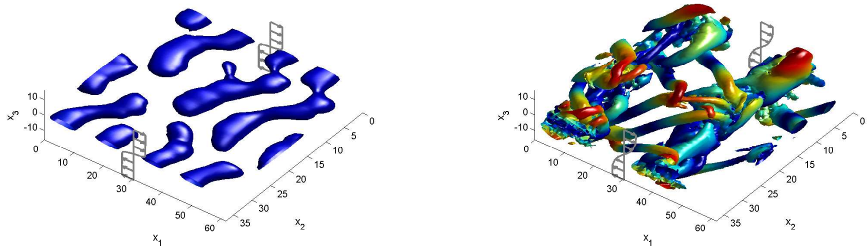
The optimisation was tested in three steps. First, the convergence of the adjoint-based gradient to the forward calculated gradient was shown. Secondly the two methods to impose the non-linear constraint were compared: the gradient-



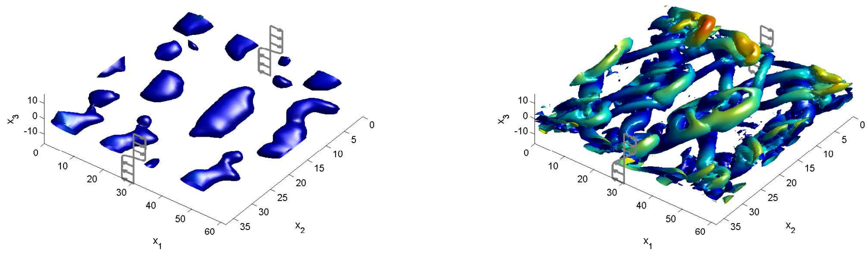
(a) Unoptimised



(b) Turbulent kinetic energy $\mathscr{J}_{TKE}$ (momentum thickness $\mathscr{J}_M$ and mean-flow kinetic energy $\mathscr{J}_{MFE}$ are similar)

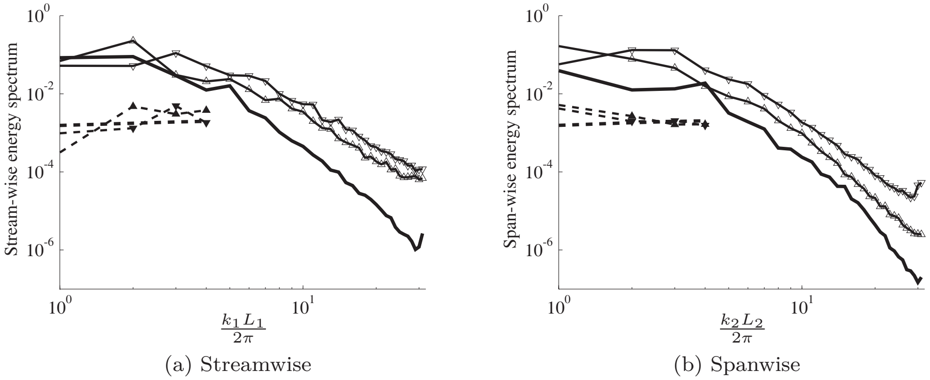


(c) Total kinetic energy $\mathscr{J}_{KE}$



(d) Enstrophy $\mathscr{J}_{ENS}$

Figure 4.8: λ_2 visualisation of the velocity-field coloured with the distance from the mixing centre line for parameters optimised to the different cost functionals with $T = 40$; profile $\langle u_1 \rangle$ (—); visualisation of $t = 0$ (left) and of $t = T$ (right)



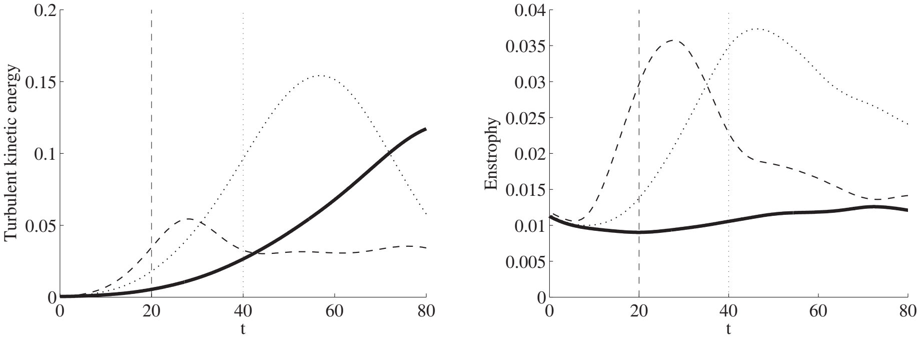
(a) Streamwise

(b) Spanwise

Figure 4.9: Energy spectra of the flow on location $x_3 = 0$ at $t = 0$ (−−) and at $t = 40$ (—); unoptimised flow (no symbol), flow optimised for $T = 40$ towards total kinetic energy (△), and enstrophy (▽); (a): streamwise energy spectrum; (b) spanwise energy spectrum

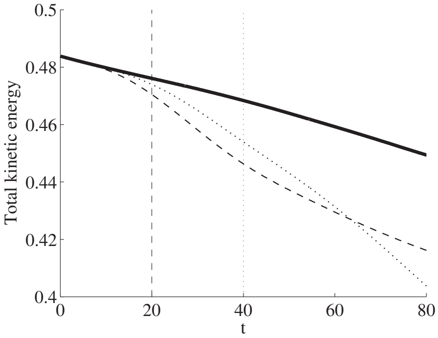
projection method, and the augmented Lagrangian method. We found that optimisation with the gradient projection method is the more robust methodology. The augmented Lagrangian method was found to be very sensitive to small gradient inconsistencies which originate from the adjoint-based gradient calculation. As a result, convergence was not always guaranteed. Finally, optimisation results for five different cost functionals, and two different time horizons were presented. The cost functionals are based on: momentum thickness, mean-flow kinetic energy, turbulent kinetic energy, total kinetic energy, and enstrophy at the time horizon. The first three of them were shown to lead to the same optima. They resulted in large two-dimensional vortex structures with maximum impact on mean momentum, but low dissipation. These structures are similar to the structures that develop starting from a combination of 2D LST modes, but the mixing layer with the optimised modes grows faster since the optimisation takes into account the non-linear interactions between the different modes. In contrast, the other two cost-functionals, maximising the enstrophy and minimising the total kinetic energy, resulted in optimal solutions that have complex three-dimensional vortex structures at different scales.

In the next chapter the total kinetic energy and enstrophy cost functionals will be used to investigate the ability to control the mixing layer for long time horizons (cfr. Aims and objectives §1.3). At long time horizons mixing layers go to self-similar behaviour, characterised by a broad spectrum of scales. Therefore we consider the enstrophy and the kinetic energy cost functionals the most interesting for the optimisation to long time horizons.



(a) Turbulent kinetic energy $\mathscr{J}_{TKE}$

(b) Enstrophy $\mathscr{J}_{ENS}$



(c) Total kinetic energy $\mathscr{J}_{KE}$

Figure 4.10: Evolution of mixing measures in time of the unoptimised flow (—) and of the optimised flow for $T = 20$ ($--$) and for $T = 40$ ($\cdots$); optimised towards turbulent kinetic energy (a), enstrophy (b) and kinetic energy (c).

We will no longer use the first three cost functionals (momentum thickness, turbulent kinetic energy and mean-flow kinetic energy), as we expect to find also large scale 2D rollers for longer time horizons. Also, the natural development of a mixing layer with such 2D rollers is the follow-up of several amalgamations. In order to facilitate several amalgamations during $[0, T]$ with T large, a large box is required with size probably at least 16 times the most unstable wavelength following from LST. Using the same cell-spacing as in 4.3.2, this would require a $512 \times 512 \times 1024$ grid. Since the optimisation requires $\mathcal{O}(10^3)$ flow simulations, this is computationally too demanding.

Chapter 5

Optimisation of a temporal mixing layer with long time horizons

In this chapter the cost functionals maximising the enstrophy and minimising the kinetic energy are used to study to which extent the mixing layer can be optimised up to long time horizons. These cost functionals are both related to the dissipation of kinetic energy. They maximise respectively the rate of dissipation at the time horizon ($\mathscr{J}_{ENS}$) and the total dissipation in the time interval $[0, T]$ ($\mathscr{J}_{KE}$). This is outlined in section 5.1.1. The properties of the box and the grid are described in section 5.1.2. In the previous chapter the optimisation of controls in the temporal mixing layer showed to be strongly non-convex, such that local optima may exist. In some cases, optimal perturbations for $T_1 < T_2$ are found to lead to lower cost functionals at $t = T_2$ than perturbations optimised for T_2. To avoid this type of situation, we adapt the procedure to select the starting point, using a simple continuation procedure in the space of optimal solutions as detailed in section 5.2. Finally the evolutions of the mixing layer with different optimal parameters are shown in section 5.3 and section 5.4. The optimised control is effective up to the longest tested time horizon, but the optimised perturbations change the moment of the onset to self-similarity compared to the unoptimised perturbations. In section 5.3, it is observed that mixing layers with perturbations that maximise the total dissipation evolve to a self-similar state almost immediately after the time horizon. The perturbations that maximise the dissipation rate on the other hand, delay the onset of self-similarity for long time horizons (section 5.4). The work discussed in this chapter is partially published in Delport, Baelmans, and Meyers (2010a,b).

5.1 Optimisation problem & case set-up

5.1.1 Optimisation problem

This chapter focusses on two cost functionals. The first cost functional minimises the total kinetic energy at the time horizon ($\mathscr{J}_{KE}$, (4.9)). Minimising $\mathscr{J}_{KE}$ corresponds with maximising the total dissipated kinetic energy up to the time horizon:

$$\begin{aligned}\mathscr{J}_{KE} &= \frac{1}{2}\frac{1}{\Omega}\int_{\Omega} \boldsymbol{u}(\mathbf{x},T,\boldsymbol{\phi}) \cdot \boldsymbol{u}(\mathbf{x},T,\boldsymbol{\phi})\,\mathrm{d}\mathbf{x} \\ &= (1+E_0)\frac{1}{2} - \int_0^T \varepsilon(T,\boldsymbol{\phi})\,\mathrm{d}t. \end{aligned} \tag{5.1}$$

with ε the volume-averaged dissipation rate.

The second cost functional maximises the enstrophy at the time horizon ($\mathscr{J}_{ENS}$, (4.12)). This is equal to maximisation of the dissipation rate $\varepsilon = 2\nu\mathscr{E}$ at the time horizon T, with $\mathscr{E}$ the volume-averaged enstrophy defined as

$$\mathscr{E}(T,\boldsymbol{\phi}) = \frac{1}{2\Omega}\int_{\Omega} \boldsymbol{\omega}(\mathbf{x},T,\boldsymbol{\phi}) \cdot \boldsymbol{\omega}(\mathbf{x},T,\boldsymbol{\phi})\,\mathrm{d}\mathbf{x}, \tag{5.2}$$

and $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$.

For sake of convenience, the optimal perturbations for a time horizon T are denoted as $\boldsymbol{\phi}_T$ (e.g., for $T = 60$, we write $\boldsymbol{\phi}_{60}$). The optimal perturbations are evaluated based on the mixing-layer evolution of total dissipated energy in time, defined as

$$\Delta E(t,\boldsymbol{\phi}_T) = \frac{\nu}{\Omega}\int_0^t \int_{\Omega} \boldsymbol{\omega}(\mathbf{x},t',\boldsymbol{\phi}) \cdot \boldsymbol{\omega}(\mathbf{x},t',\boldsymbol{\phi})\,\mathrm{d}\mathbf{x}\,\mathrm{d}t'. \tag{5.3}$$

Likewise we will monitor the dissipation rate, by evaluating the volume-averaged enstrophy in time, i.e. $\mathscr{E}(t,\boldsymbol{\phi}_T)$ using its definition (5.2). It is clear that these properties are related to the cost functionals as $\Delta E(T,\boldsymbol{\phi}_T) = (1+E_0)(1/2) - \mathscr{J}_{KE}$ and $\mathscr{E}(T,\boldsymbol{\phi}_T) = -\mathscr{J}_{ENS}$.

Section 1.2 explained that mixing layers evolve to self-similar states, in which the properties of the mixing layer remain constant if scaled appropriately. For the temporal mixing layer the dissipation rate is constant in time in the self-similar state, while the total dissipated energy evolves linear in time. If there is only one unique self similar state, then this behaviour can make it impossible to enhance the dissipation properties of the mixing layer at long time horizons, corresponding to a long distance behind the splitter plate in the spatial mixing layer. However, opposite views exist on the question if there are only one or several self-similar states. Some authors argue that flows “remember forever” how they began (George, 1989; Balaras et al., 2001), others attain the view that ‘different’ self-similar states are only observed in experiments and simulations because shear flows only “slowly forget” how they began, arguing that these differences are the effect of a slow transition to a unique self-similar state (Dimotakis and Brown, 1976). This raises some questions: is it possible to ‘push’ a (temporal) mixing-layer into two distinct self-similar states via control of its initial state, or does only a one self-similar solution exist for longer time horizons? Does optimisation for long time horizons automatically lead to self-similar behaviour? Is it possible to ‘delay’ self-similarity in favour of a desired property?

This chapter tests the effectiveness of low-energy controls at long time horizons. The optimised flows will be analysed with regard to their self-similarity properties.

5.1.2 Case set-up

The box-size in streamwise and spanwise direction corresponds to $L_1 = 123.2 = 8\lambda_1$, and $L_2 = 74.04 = 8\lambda_2$, or 8 times the most unstable wavelength of the current mixing layer (cfr. 3.2.1). The box size in the normal direction L_3 is 240.

All simulations are performed on a $128 \times 128 \times 256$ mesh. In the normal direction the grid is stretched, with a grid-spacing $\Delta x_3 \sim |x_3|^{1/4}$. This particular stretching is selected based on the observation that, for self-similar mixing layers, the Kolmogorov scale $\eta \sim \theta^{1/4}$, with θ the momentum thickness of the layer.

For the optimisation in the current study, we restrict the initial perturbations $\boldsymbol{\phi}$ in stream- and spanwise directions to wavelengths larger or equal to $1/16^{\text{th}}$ of the box size. Hence, $\boldsymbol{\phi}$ contains Fourier modes with wave numbers $(k_1, k_2) = (\alpha\, 2\pi/L_1, \beta\, 2\pi/L_2)$, with $\alpha, \beta = 0, \pm 1, \cdots \pm 16$ (the mode $\alpha = \beta = 0$ is excluded). In total, this yields $\mathcal{O}(10^5)$ degrees of freedom in the controls[1], which are to be optimised.

5.2 Optimisation procedure

The gradient-based optimisation, tested in chapter 4, is used. The energy constraint is taken into account with the gradient-projection method. In this section, the selection of the starting point is further detailed. For non-convex optimisation, gradient-based procedures may lead to local optima which depend

---

[1] 2×256 values per optimised mode (with 256 the number of grid points in the normal direction and factor 2 since 2 of the 3 velocity components are free parameters) $\times$ $(33 \times 33 - 1)$ modes

on the starting point of the optimisation procedure. Here, the focus is on a reliable procedure, which tries to avoid inconsistent local optima.

In chapter 4, it was found that some optima are only local optima. For instance, when optimisation is performed to $\mathscr{J}_{KE}$, it is expected for two different optimisation time horizons that $\Delta E(T_2, \boldsymbol{\phi}_{T_1}) \leq \Delta E(T_2, \boldsymbol{\phi}_{T_2})$ (cf. section 5.1.1). However, in chapter 4 we found that this is not always the case while using the conjugate gradient procedure starting from the same initial guess for the controls (i.e., this was observed for $T_1 = 20$, $T_2 = 40$, and $\mathscr{J}_{KE}$). Also at the outset of the study described in this chapter, we were faced with this type of local optima, when starting the conjugate-gradient iterations from the same starting point $\boldsymbol{\phi}_A$.

In order to get a more consistent characterisation of optima as a function of the time horizon T, we changed the optimisation approach, as schematically shown in figure 5.1. Instead of starting all optimisations for different time horizons directly from $\boldsymbol{\phi}_A$, we only start the optimisation for the lowest time horizon ($T = 60$) from $\boldsymbol{\phi}_A$. Subsequently, the optimal parameter set $\boldsymbol{\phi}_{60}$ is used as a starting point for the optimisation to the next time horizon ($T = 80$). During the first outer iteration of the optimisation algorithm, the mixing layer is simulated with $\boldsymbol{\phi} = \boldsymbol{\phi}_{60}$ from $t = 0$ to 80, the adjoint simulation is performed from $t = 80$ to 0, the search direction is calculated, and the line search is started. After many outer iterations the conjugate gradient optimisation converges to $\boldsymbol{\phi}_{80}$. $\boldsymbol{\phi}_{80}$ is subsequently used as a starting point for the optimisation to the next time horizon and so forth until the last time horizon $T = 160$. In this way, $\Delta E(T_2, \boldsymbol{\phi}_{T_1}) \leq \Delta E(T_2, \boldsymbol{\phi}_{T_2})$ is guaranteed when $\Delta E(T, \boldsymbol{\phi}_T)$ is maximised (with T_1 and T_2 two subsequent time horizons), and $\mathscr{E}(T_2, \boldsymbol{\phi}_{T_1}) \leq \mathscr{E}(T_2, \boldsymbol{\phi}_{T_2})$ in case $\mathscr{E}(T, \boldsymbol{\phi}_T)$ is maximised. This procedure corresponds to a simple continuation approach in the space of optimal solutions. If we presume that $\boldsymbol{\phi}_T$ depends continuously on T, this is nothing more than a piecewise constant approximation of $\boldsymbol{\phi}$, which we use as starting point for the conjugate-gradient method. In the current work, higher-order continuation approaches were not further investigated.

Notwithstanding the procedure described above, the optimisation problem remains non-convex, and local optima may still exist. In particular for optimisation towards maximum rate of dissipation of kinetic energy at the time horizon ($\mathscr{J}_{ENS}$), we identified two distinct optima (cf. section 5.4 below for a further discussion on these optima). These optima were found by trying a number of different initial starting points; the two starting points $\boldsymbol{\phi}_A(\mathbf{x})$ and $\boldsymbol{\phi}_B(\mathbf{x})$ which led us to these local optima are briefly discussed here. In figure 5.2 the mixing-layer evolution starting from these two initial conditions is displayed. It is observed that these initial perturbations (using $E_0 = 10^{-4}$) lead to distinctly different evolutions of the temporal mixing layer, both when evaluating ΔE and $\mathscr{E}$, or when looking at the coherent structures using the λ_2 criterion (cfr. Jeong and Hussain (1995), and p. 66). The construction of the perturbations $\boldsymbol{\phi}_A(\mathbf{x})$ and $\boldsymbol{\phi}_B(\mathbf{x})$ themselves, is rather *ad hoc*, but is briefly added here for sake of completeness.

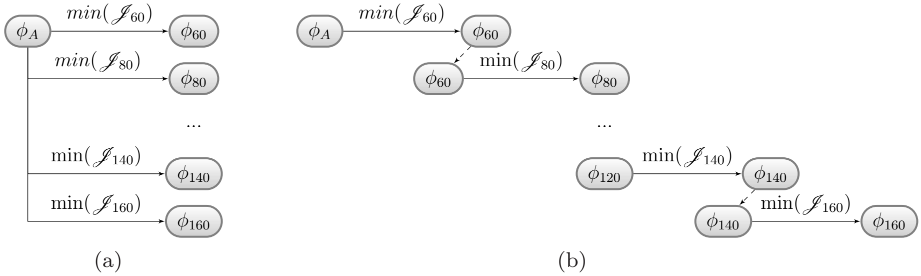


Figure 5.1: Optimisation procedure for a set of optimisations towards different time horizons: (a) as used in chapter 4, (b) currently used continuation procedure.

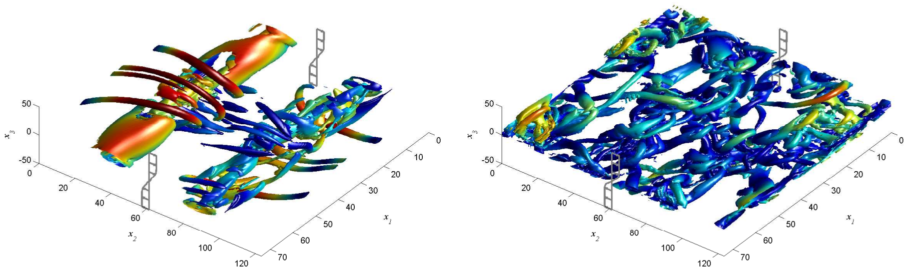
The first field, $\boldsymbol{\phi}_A(\mathbf{x})$, is constructed similarly to the initial conditions reported by Moser and Rogers (1993) and Vreman et al. (1997). The perturbations are a superposition of two-dimensional modes (having index $(\alpha, 0)$, $\alpha = 1 \cdots 16$) and three-dimensional modes (having index (α, β), $\beta \neq 0$). Fourier index (α, β) corresponds to wavenumber $(\alpha \;\, 2\pi/L_1, \beta \;\, 2\pi/L_2)$. All two-dimensional modes use the most unstable eigenfunction in the normal direction for that particular wavelength, following from a linear stability analysis. The phases of the modes are set to zero. For the three dimensional modes, a Gaussian profile (with $\sigma = 3.5$) is used in the normal direction for u_2 and u_3 (the profile for u_1 follows from u_2 and u_3 using the continuity equation). The total energy is evenly distributed over all modes.

The second set of initial perturbations, $\boldsymbol{\phi}_B(\mathbf{x})$, is composed of broadband random noise for 20% of its kinetic energy, and of low-pass-filtered broadband random noise for the remaining 80% of its total kinetic energy. The low-pass filter employed is a Gaussian filter. Its width is selected such that the transfer function of the filter has a value of 0.5 at the most unstable LST wavenumber $(8 \;(2\pi/L_1), 0)$.

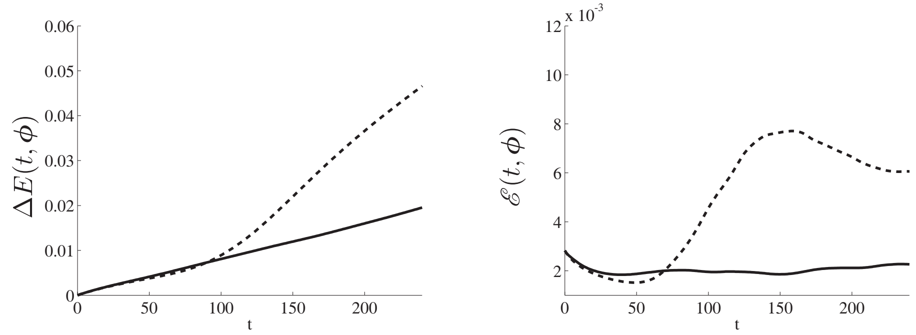
5.3 Maximise total dissipated energy

5.3.1 Optimisation results

Optimisation results for the maximisation of dissipated total kinetic energy, optimised with cost functional $\mathscr{J}_{KE}$, are now presented in figure 5.3 for two energy levels $E_0 = 10^{-5}$ and 10^{-4}. The optimisation started from $\boldsymbol{\phi}_A$, and used the recursive procedure described above. In figure 5.3 (a) and (c) the convergence history as function of the number of conjugate-gradient (outer) iterations is



(a) ϕ_A (b) ϕ_B

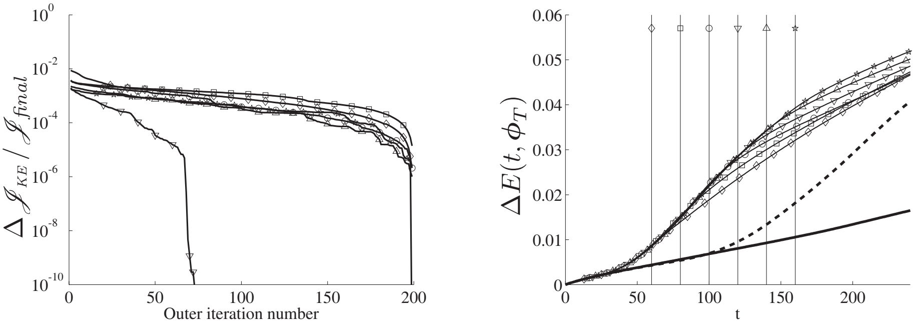


(c) Evolution of ΔE (d) Evolution of $\mathscr{E}$

Figure 5.2: Mixing-layer evolution starting from two different initial perturbations ϕ_A and ϕ_B (with $E_0 = 10^{-4}$). (a,b) λ_2 visualisation Jeong and Hussain (1995) of the velocity field at time $t = 120$, coloured with the distance from the mixing centre plane; (— (in grey)): profiles for $\langle u_1 \rangle$. (c) Evolution of the dissipated energy ΔE and (d) Enstrophy $\mathscr{E}$ for initial conditions (—) ϕ_A and ($--$) ϕ_B.

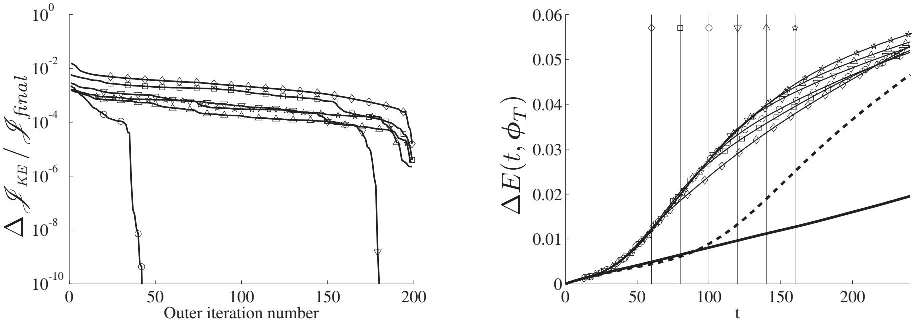
presented. The convergence criterion is based on a relative improvement of the cost functional of 10^{-10}. Some of the maximisations of dissipated total kinetic energy are not converged up to this criterion, but were stopped instead after 200 iterations for resource reasons. We recall that one conjugate gradient (outer) iteration requires one adjoint Navier–Stokes simulation for the gradient calculation, and about 8 Navier–Stokes simulations for the line-search algorithm. One Navier–Stokes simulation of the mixing layer on the current $128 \times 128 \times 256$ grid, up to time horizon $T = 160$, takes about one hour when 32 CPU's are used.

The evolution of the total dissipated energy $\Delta E(t, \phi_T)$ is displayed for different time horizons in figure 5.3 (b) and (d), respectively for $E_0 = 10^{-5}$ and 10^{-4}. A large increase of the total dissipated energy for the optimised cases is seen when



(a) $E_0 = 10^{-5}$, starting point $\boldsymbol{\phi}_A$

(b) $E_0 = 10^{-5}$, starting point $\boldsymbol{\phi}_A$



(c) $E_0 = 10^{-4}$, starting point $\boldsymbol{\phi}_A$

(d) $E_0 = 10^{-4}$, starting point $\boldsymbol{\phi}_A$

Figure 5.3: Optimisation results for $\mathscr{J}_{KE}$ starting from $\boldsymbol{\phi}_A$. (a,c) Convergence history for the different time horizons (with $\Delta \mathscr{J}_{KE} = \mathscr{J}_{KE}(it) - \mathscr{J}_{final}$). (b,d) Evolution of the total dissipated energy $\Delta E(t, \boldsymbol{\phi}_T)$ as function of time for different time horizons. Symbols in (a,b,c,d): $T = 60$ ($\Diamond$), $T = 80$ ($\Box$), $T = 100$ ($\circ$), $T = 120$ ($\triangledown$), $T = 140$ ($\vartriangle$), $T = 160$ ($\star$). Additional lines in (b,d): Evolution of the total dissipated energy for initial condition $\boldsymbol{\phi}_A$ (—) and $\boldsymbol{\phi}_B$ (−−); vertical lines are drawn at the different optimisation time horizons.

compared to the evolution with $\boldsymbol{\phi}_A$ (also displayed in the figures). We observe that, for $T_2 > T_1$, the corresponding optimal energy levels $\Delta E(T_1, \boldsymbol{\phi}_{T_2}) \approx \Delta E(T_1, \boldsymbol{\phi}_{T_1})$. Hence, for the time horizons investigated here, the optimal solution for a time horizon T is also optimal for all time horizons smaller than T.

To further analyse this, it is useful to focus on the maximum dissipated total kinetic energy for each time horizon $\Delta E(T, \boldsymbol{\phi}_T)$ as function of the time horizon T (instead of $\Delta E(t, \boldsymbol{\phi}_T)$). $\Delta E(T, \boldsymbol{\phi}_T)$ is based on the optima of the optimisations to the different time horizons $T = [60, 80, \ldots, 160]$, and depicts the evolution

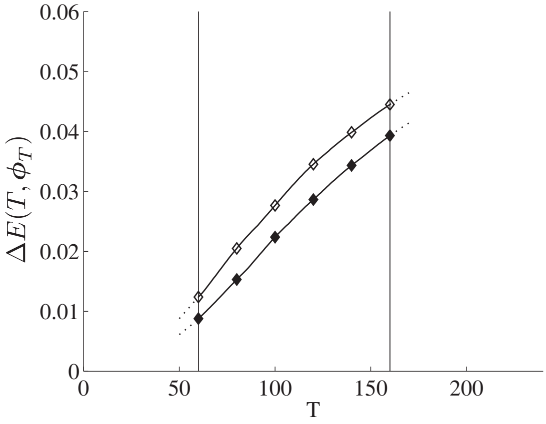


Figure 5.4: Maximal total dissipated energy as function of time horizon T for flows optimised towards $\mathscr{J}_{KE}$. (—♦) $E = 10^{-5}$, and (—◊) $E = 10^{-4}$.

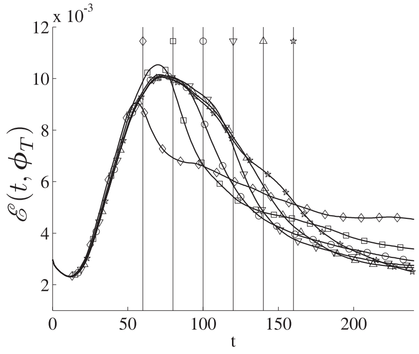


Figure 5.5: Evolution of the enstrophy in time for flows optimised towards $\mathscr{J}_{KE}$ with $T = 60$ (◊), $T = 80$ (□), $T = 100$ (○), $T = 120$ (▽), $T = 140$ (△), $T = 160$ (⋆) for $E_0 = 10^{-4}$.

of the total dissipated energy as function of the time horizon. Figure 5.4 shows $\Delta E(T, \boldsymbol{\phi}_T)$ as a monotonic function that increases with T. We find that $\Delta E(T, \boldsymbol{\phi}_T) = \Delta E(T, \boldsymbol{\phi}_{160})$, with $T = 160$ the longest optimisation time horizon included in the current study. $\Delta E(t, \boldsymbol{\phi}_T)$ can also be expressed as

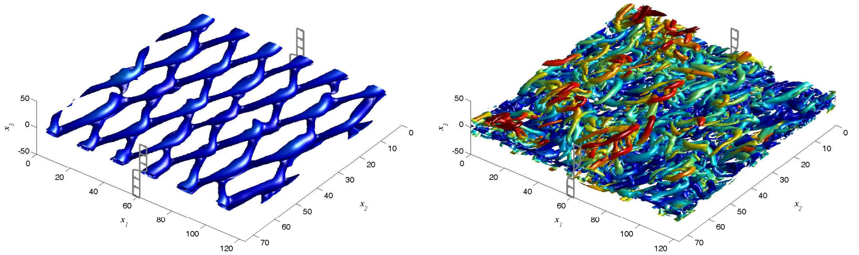
$$\Delta E(t, \boldsymbol{\phi}_T) \quad = \quad 2\nu \int_0^t \mathscr{E}(t', \boldsymbol{\phi}_T)\,\mathrm{d}t', \tag{5.4}$$

with $\mathscr{E}(t, \boldsymbol{\phi}_T)$ the volume-averaged enstrophy evolution starting from the $\mathscr{J}_{KE}$ -optimal perturbations $\boldsymbol{\phi}_T$. Therefore, $\Delta E(T_1, \boldsymbol{\phi}_{T_2}) \approx \Delta E(T_1, \boldsymbol{\phi}_{T_1})$ for $T_2 > T_1$, requires that the time integrals from 0 to T_1 of $\mathscr{E}(t, \boldsymbol{\phi}_{T_1})$ and $\mathscr{E}(t, \boldsymbol{\phi}_{T_2})$ are equal. This is illustrated in fig. 5.5, where the enstrophy evolution of flows optimised to $\mathscr{J}_{KE}$ (and $E_0 = 10^{-4}$) is shown. The maximisation of the dissipated kinetic energy corresponds with maximisation of the time integral of the enstrophy. This results in an early peak of the enstrophy and an high enstrophy value is sustained as long as possible afterwards until the time horizon.

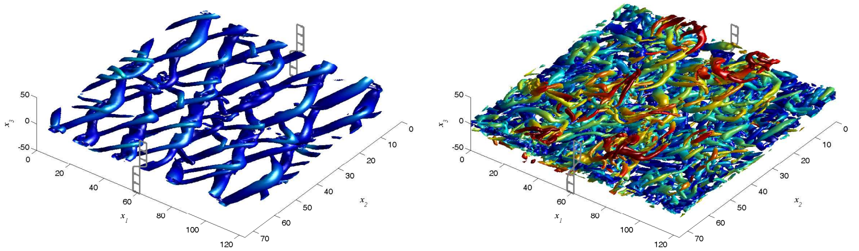
Moreover, comparing $\Delta E(T, \boldsymbol{\phi}_T)$ for $E_0 = 10^{-5}$ and $E_0 = 10^{-4}$ in fig. 5.4, we observe a straightforward effect of the control energy: more energy in the initial perturbations yields higher dissipated energy for all optimisation time horizons.

The flow structures of the mixing layer optimised to $\mathscr{J}_{KE}$ with time horizon $T = 160$ are visualised in figure 5.6 using the λ_2 criterion (cfr. Jeong and Hussain (1995), and p. 66). The structures observed for the other time horizons are very similar, and are not further shown here. In figure 5.6 (right) the flow is visualised at the time horizon. It shows a fully turbulent mixing layer, dominated by small-scale structures. The Taylor-Reynolds number measured in the mixing centre plane is $Re_\lambda \approx 40$ at this point in time. It is observed in figure 5.6 (left) that the coherent structures at $t = 30$ form a diamond-shaped vortex pattern. The diamonds originate from a high initial energy concentration around the $(6, \mp 2)$ mode (see fig. 5.7). This mode has as wavelength $(8/6\lambda_1, 8/2\lambda_2)$ (with $\lambda_1 = 15.4$ and $\lambda_2 = 0.6\lambda_1$ cf. section 5.1.2). Figure 5.7 depicts the energy of $\boldsymbol{\phi}_{160}$ per mode E_{mode}. It shows that a limited number of modes have a high initial energy level. When the control energy is redistributed to the n modes with the highest energy level in the optimised control for $E_0 = 10^{-5}$, it is found that the total dissipation reaches 66% of the optimal dissipation for $n = 2$ (only mode $(6, 2)$ and $(6, -2)$). For $n = 4$, the total energy dissipated corresponds to 74% of the optimal value. For $n = 8$, a total dissipation is found that is only slightly higher (75%). Hence, using only 4 modes, up to 75% of the energy dissipation realised by optimal control, may be recovered.

Similar diamond-shaped vortex structures have also been observed in an experiment of a forced mixing layer by Nygaard and Glezer (Fig. 5 in Nygaard and Glezer (1990)). They tested the impact of a spanwise array of surface heaters, placed on a splitter plate, on the subsequent flow evolution. With an adequate phase difference



(a) $E_0 = 10^{-5}$, $T = 160$



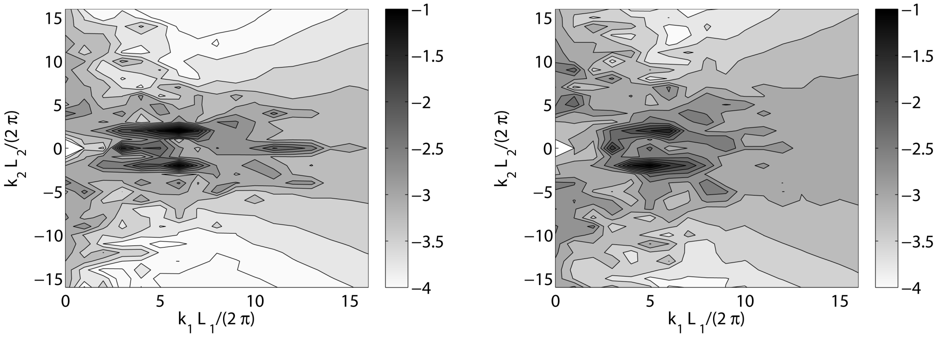
(b) $E_0 = 10^{-4}$, $T = 160$

Figure 5.6: λ_2 visualisation of the velocity-field coloured with the distance from the mixing centre line for parameters optimised towards $\mathscr{J}_{KE}$; profile $\langle u_1 \rangle$ (—); visualisation of $t = 30$ (left) and $t = 160$ (right)

between the heaters, diamond-shaped structures are obtained. Collis et al. (1994) showed that these structures may also be observed in an incompressible temporal mixing layer when a pair of oblique disturbances is used for the initial condition, which is similar to our observation. They also showed that this flow behaviour increases mixing compared to standard rib/roller flows (Collis et al., 1994).

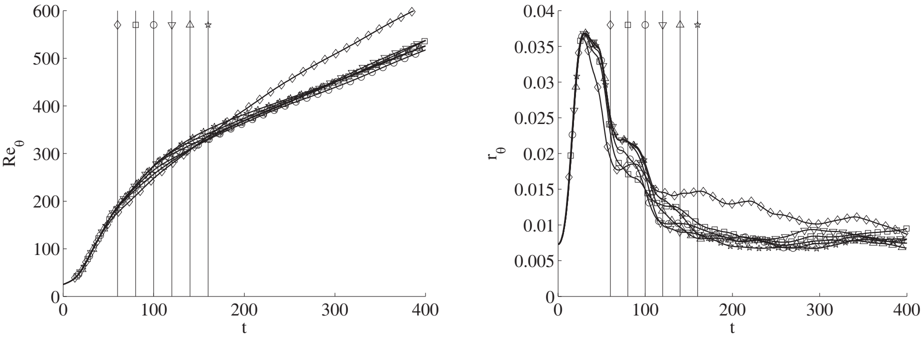
5.3.2 Self-similarity

The evolution of optimised mixing layers is now studied beyond the time horizon for which the controls are optimised. It is observed that all cases reach a self-similar mixing-layer evolution. For temporal mixing layers, self-similar states are characterised by a linear growth of the momentum thickness (4.7) in time. Also, the turbulent kinetic energy grows linearly in time, and the ratio between the total production and total dissipation of turbulent kinetic energy is constant. Finally, small turbulent scales in the flow are in equilibrium, which can, e.g. be verified by looking at the balance between enstrophy production and dissipation. Various



(a) $E_0 = 10^{-5}$, $T = 160$ (b) $E_0 = 10^{-4}$, $T = 160$

Figure 5.7: Energy distribution over the optimised modes of the optimised initial perturbations, coloured by $\log(E_{mode}/(E_0/2))$. Energy is integrated over x_3

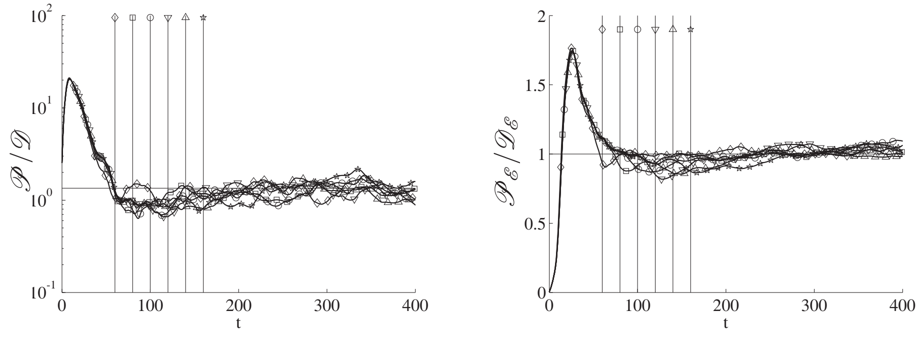


(a) Re_θ (b) r_θ

Figure 5.8: Evolution of (a) the Reynolds number based on the momentum thickness Re_θ and (b) the momentum thickness growth rate in time for flows optimised towards $\mathscr{J}_{KE}$, $E_0 = 10^{-4}$. $T = 60$ $(\Diamond)$, $T = 80$ $(\Box)$, $T = 100$ $(\circ)$, $T = 120$ $(\triangledown)$, $T = 140$ $(\triangle)$, $T = 160$ $(\star)$

other diagnostics can be investigated, such as the collapse of mean velocity profiles, and Reynolds stresses when appropriately scaled with θ and ΔU, etc. More details are, e.g., found in Rogers and Moser (1994); Moser, Rogers, and Ewing (1998); Balaras et al. (2001); Pope (2000).

In figure 5.8 (a) the Reynolds number based on the momentum thickness is



(a) $\mathscr{P}/\varepsilon$ (b) $\mathscr{P}_{\mathscr{E}}/\mathscr{D}_{\mathscr{E}}$

Figure 5.9: (a) Evolution of $\mathscr{P}/\mathscr{D}$ in the centre of the layer in time. Horizontal line at 1.35 (b) Evolution of the rate of enstrophy production to enstrophy dissipation in time. Horizontal line at 1.0. (a,b) Flows optimised towards $\mathscr{J}_{KE}$ with $T = 60$ ($\Diamond$), $T = 80$ ($\Box$), $T = 100$ ($\circ$), $T = 120$ ($\triangledown$), $T = 140$ ($\triangle$), $T = 160$ ($\star$), for $E_0 = 10^{-4}$.

displayed for flow optimised to $\mathscr{J}_{KE}$ with $E_0 = 10^{-4}$. It is observed that Re_θ starts to evolve linearly in time, corresponding with linear evolution of θ. This is a typical property of a self-similar mixing layer. The growth rate $r_\theta \equiv \mathrm{d}\theta/\mathrm{d}t/\Delta U$ of the mixing-layer momentum thickness, given in figure 5.8 (b), confirms the linear evolution of θ and thus Re_θ. The figure shows that the optimised mixing layers roughly reach a constant growth rate for $t \geq 160$, with values between 0.007 and 0.012. This range is a bit lower than the value of 0.014 obtained in the study of Rogers and Moser (1994).

In figure 5.9 (a), the ratio between production and dissipation of turbulent kinetic energy at the mixing-layer centre is shown. They are respectively defined as (see, e.g. Pope (2000))

$$\mathscr{P} = -\frac{1}{L_1 L_2} \int \left(u_1' u_3' \frac{\partial \langle u_1 \rangle}{\partial x_3} \right) \bigg|_{x_3=0} \mathrm{d}x_1 \, \mathrm{d}x_2, \tag{5.5}$$

and

$$\mathscr{D} = \frac{1}{L_1 L_2} \int 2\nu \, (\boldsymbol{s} : \boldsymbol{s})_{x_3=0} \, \mathrm{d}x_1 \, \mathrm{d}x_2, \tag{5.6}$$

with $\boldsymbol{s} = [\nabla \boldsymbol{u}' + (\nabla \boldsymbol{u}')^T]/2$ the shear-stress tensor related to the fluctuating velocity $\boldsymbol{u}' = \boldsymbol{u} - \langle \boldsymbol{u} \rangle$ (and $\langle \boldsymbol{u} \rangle$ the velocity averaged over x_1–x_2 planes). For all cases, we observe a ratio $\mathscr{P}/\mathscr{D} \approx 1.35$, evaluated in the mixing layer's self-similar region. This corresponds well to the ratio of 1.4 which was observed by Rogers and Moser (1994).

To investigate the equilibrium of the small-scale turbulence, the ratio of enstrophy production to enstrophy dissipation is displayed in figure 5.9 (b). Production and dissipation of enstrophy are defined by (cf. Pope (2000)):

$$\mathscr{P}_{\mathscr{E}} = \frac{1}{\Omega} \int_{\Omega} (\boldsymbol{\omega} \cdot \nabla \boldsymbol{u}) \cdot \boldsymbol{\omega} \, \mathrm{d}\mathbf{x}, \tag{5.7}$$

and

$$\mathscr{D}_{\mathscr{E}} = \frac{1}{\Omega} \int_{\Omega} \nu (\nabla \boldsymbol{\omega} : \nabla \boldsymbol{\omega}) \, \mathrm{d}\mathbf{x}. \tag{5.8}$$

A ratio of enstrophy production to enstrophy dissipation of 1.0 is reached in all cases for $t > 200$–240. In this case production and dissipation of enstrophy are balanced, and we observe (not shown here) a constant enstrophy level of 3.7×10^{-3} for all cases; this value is closely corresponding to the values reported in Balaras et al. (2001) and Rogers and Moser (1994). The Taylor-Reynolds number in our optimised cases, measured in the mixing centre plane, ranges between $Re_\lambda \approx 40$ (at $t = 160$) and $Re_\lambda \approx 55$ (at $t = 400$).

5.3.3 Discussion

The results in figures 5.8–5.9 illustrate self-similar mixing evolution for all optimised cases when the simulations are sufficiently continued beyond the optimisation time horizon. The Reynolds number based on momentum thickness $Re_\theta \equiv \theta \Delta U / \nu$ at which a self-similar state emerges is significantly lower than values reported in earlier studies. We find in our optimised cases $Re_\theta \approx 350$ at the onset of self-similarity (evaluated at $t = 200$, cfr. fig. 5.8 (a)), which should be compared to $Re_\theta \approx 1900$ found by Balaras et al. (2001) and $Re_\theta \approx 1440$ found by Rogers and Moser (1994). Hence, in the current thesis, Reynolds numbers at the start of the self-similar region are factor of 4 to 5 lower. The Taylor-Reynolds number at the onset of self-similarity in our study corresponds to $Re_\lambda \approx 40$. Hence, compared to uncontrolled initial conditions, optimisation to $\Delta E(T, \boldsymbol{\phi}_T)$ leads to a significant speed up of the transition of the mixing layer to a self-similar state, which is reached at much lower Reynolds numbers.

All flows optimised to $\Delta E(T, \boldsymbol{\phi}_T)$ have a maximum dissipation rate ($\varepsilon = 2\nu\mathscr{E}$ and fig. 5.5) well before the onset of self-similarity during the early transition of the mixing layer (around $t = 70$). In the next section it is attempted to maximise the dissipation rate at the time horizon T, for time horizons up to $T = 160$.

5.4 Maximise dissipation rate

5.4.1 Optimisation results

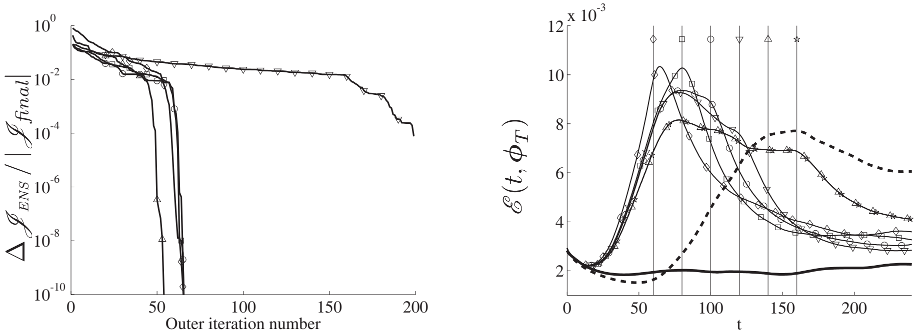
This section focuses on the maximisation of the energy dissipation-rate at the time horizon, which corresponds with the maximisation of enstrophy at $t = T$. The optimisation is performed for energy levels $E_0 = 10^{-4}$ and $E_0 = 10^{-5}$ starting from point $\boldsymbol{\phi}_A$ and using the procedure to select the starting points described in figure 5.1 (b). Next to that, for one case ($\mathscr{J}_{ENS}$ and $E_0 = 10^{-4}$), the iterative procedure is started from a different starting point $\boldsymbol{\phi}_B$, again using six different time horizons.

Optimisation results for $\mathscr{J}_{ENS}$ and $E_0 = 10^{-4}$ starting from $\boldsymbol{\phi}_A$ are presented in figure 5.10. In figure 5.10 (a) the convergence history as function of the number of conjugate-gradient (outer) iterations is presented. We used again the convergence criterion based on a relative improvement of the cost functional of 10^{-10}. It is shown that optimisations to all time horizons converge well with respect to this criterion, except for the optimisation to $T = 120$, which was stopped after 200 iterations for computational resource reasons.

In figure 5.10 (b) the enstrophy evolution $\mathscr{E}(t, \boldsymbol{\phi}_T)$ is displayed for the different time horizons. A large increase of the enstrophy for the optimised cases is obtained when compared to the enstrophy evolution of $\boldsymbol{\phi}_A$ (also displayed in the figure). For low time horizons ($T = 60$ and 80), the enstrophy evolution strongly peaks near the time horizon for which it is optimised, followed by a sharp decline of enstrophy for $t > T$. For longer time horizons ($T \geq 100$), this is no longer the case. Now peaks around $t = 80$ are observed followed with a gradual decline up to the time horizon (t=T). For $t > T$, again a sharper decline in enstrophy is seen (note that, for the current case, the optimal solutions found for $T = 140$ and $T = 160$ are the same).

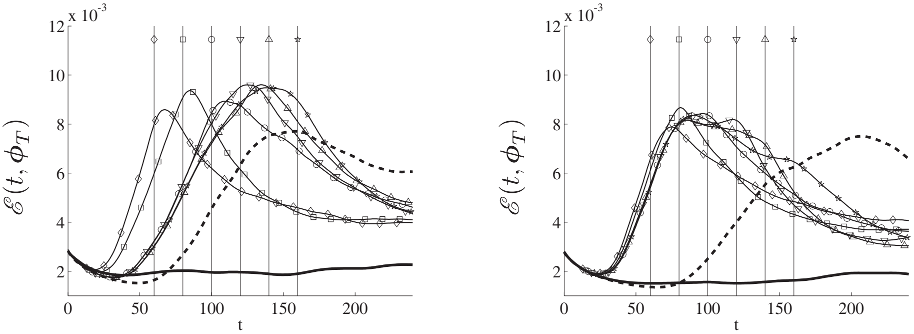
In figure 5.10 (b), the evolution of the enstrophy starting from the initial field $\boldsymbol{\phi}_B$ (without any optimisation) is also displayed. This (unoptimised) solution yields a higher level of enstrophy than the optimal solutions $\mathscr{E}(t, T)$ for $T > 120$ optimised starting from $\boldsymbol{\phi}_A$. Consequently, for $T > 120$, these optimised solutions are local optima. To further investigate this, optimisation starting from point $\boldsymbol{\phi}_B$ is considered. For the remainder of the work, we will refer to optima coming from the initial optimisation starting point $\boldsymbol{\phi}_A$ as ‘group A’, and similarly, for $\boldsymbol{\phi}_B$, ‘group B’.

In figure 5.11 (a), the evolution of the enstrophy is shown using $E_0 = 10^{-4}$ for the optimised flows of group B. For the optimisation to the different time horizons of group B, we slightly changed the optimisation sequence used for group A (shown in figure 5.1 (b)). The procedure was changed based on figure 5.10 (b). This



(a) (b)

Figure 5.10: Optimisation results for $\mathscr{J}_{ENS}$ and $E_0 = 10^{-4}$ starting from $\boldsymbol{\phi}_A$. (a) Convergence history for the different time horizons (with $\Delta \mathscr{J}_{ENS} = \mathscr{J}_{ENS}(it) - \mathscr{J}_{final}$). (b) Evolution of the enstrophy $\mathscr{E}(t, \boldsymbol{\phi}_T)$ as function of time for different time horizons. Symbols in (a,b): $T = 60$ ($\Diamond$), $T = 80$ ($\Box$), $T = 100$ ($\circ$), $T = 120$ ($\triangledown$), $T = 140$ ($\vartriangle$), $T = 160$ ($\star$). Additional lines in (b): enstrophy evolution of initial condition $\boldsymbol{\phi}_A$ (—) and $\boldsymbol{\phi}_B$ (−−); vertical lines are drawn at the different optimisation time horizons.



(a) $E_0 = 10^{-4}$, starting point $\boldsymbol{\phi}_B$

(b) $E_0 = 10^{-5}$, starting point $\boldsymbol{\phi}_A$

Figure 5.11: Evolution of the enstrophy in time for flows optimised towards $\mathscr{J}_{ENS}$ and $T = 60$ ($\Diamond$), $T = 80$ ($\Box$), $T = 100$ ($\circ$), $T = 120$ ($\triangledown$), $T = 140$ ($\vartriangle$), $T = 160$ ($\star$); also shown: enstrophy evolution of initial condition $\boldsymbol{\phi}_A$ (—) and $\boldsymbol{\phi}_B$ (−−).

figure shows that $T = 140$ is the lowest time horizon for which the enstrophy with unoptimised initial condition $\boldsymbol{\phi}_B$ is higher than the optimised enstrophy $\mathscr{E}(T, \boldsymbol{\phi}_T)$ from group A. Therefore, we first initialise the conjugate-gradient procedure with starting point $\boldsymbol{\phi}_B$ for an optimisation towards time horizon $T = 140$. Subsequently, the optimal controls for $T = 140$ are used as starting point towards $T = 160$, and $T = 120$. The optimal control for $T = 120$ is used as initial points for further decrease of the time horizon, etc. The optimisations for all time horizons converge well up to the relative convergence criterion of 10^{-10}. These convergence results are not further shown here. As is depicted in figure. 5.11 (a), the optimisation results based on $\boldsymbol{\phi}_B$ provide a large increase of enstrophy compared to the unoptimised $\boldsymbol{\phi}_B$ simulation.

In figure 5.11 (b) results with respect to $\mathscr{J}_{ENS}$ are also presented for the lower energy level $E_0 = 10^{-5}$, using the starting point $\boldsymbol{\phi}_A$ (also used for figure 5.10 (b)). The optimisation procedure shown in figure 5.1 (b) is used and convergence for all time horizons up to the relative convergence criterion of 10^{-10} is reached (not shown here). It is seen that also for a lower energy level of the controls, a significant improvement of the enstrophy at the optimisation time horizon can be obtained compared to the unoptimised starting point $\boldsymbol{\phi}_A$.

We now analyse to what extent the total enstrophy in a mixing layer can be maximised for long time horizons. The maximum enstrophy for each time horizon $\mathscr{E}(T, \boldsymbol{\phi}_T)$ is studied as a function of the time horizon T (instead of $\mathscr{E}(t, \boldsymbol{\phi}_T)$). Next to that, we investigate to what type of spatial structures the optimal perturbations $\boldsymbol{\phi}$ lead.

In figure 5.12, the optimal enstrophy $\mathscr{E}(T, \boldsymbol{\phi}_T)$ is presented as function of the time horizon for optimisation with $E_0 = 10^{-4}$ (group A, and B), and with $E_0 = 10^{-5}$ (group A). Focussing first on optimisation using an energy level $E_0 = 10^{-4}$ for the controls, a difference between optima in group A and group B is seen. For $T < 100$, optimisations in group A yield the highest enstrophy $\mathscr{E}(T, \boldsymbol{\phi}_T)$, while the inverse holds for $T > 100$. Overall, combining the best optima of group A and group B, we find that optimal controls lead to a maximal enstrophy larger than 8.6×10^{-3}, which is almost three times higher than the eventual equilibrium level in the self-similar region (cf. section 5.4.2 for further discussion). Using the best optima from group A and B, a slight drop in the level of maximal enstrophy (about 10%) is observed for longer time horizons compared to shorter optimisation time horizons.

In figure 5.12, the maximal enstrophy $\mathscr{E}(T, \boldsymbol{\phi}_T)$ of cases with control energy level $E_0 = 10^{-5}$ and $E_0 = 10^{-4}$ is also compared (both using the same optimisation starting point $\boldsymbol{\phi}_A$ at a different energy level). For short time horizons ($T < 100$), the higher energy level realises a higher level of enstrophy. For longer time horizons, this difference disappears, and enstrophy levels for both cases are roughly equal. This illustrates that the effect of the energy level is not linearly transferred to

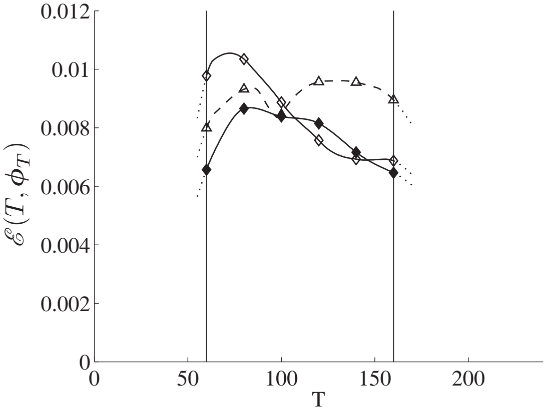
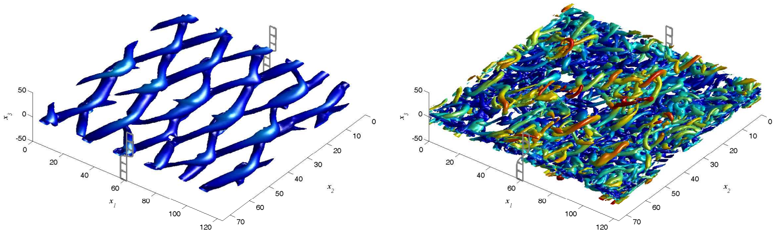


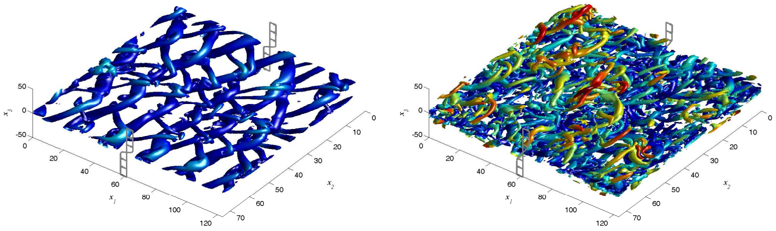
Figure 5.12: Maximal enstrophy as function of time horizon T for flows optimised towards $\mathscr{J}_{ENS}$. (—◊) group A, $E_0 = 10^{-4}$; (−−△) group B, $E = 10^{-4}$; and (—◆) group A, $E = 10^{-5}$.

the optimal values of the cost functionals. Especially for higher time horizons, it appears interesting to investigate the (in)effectiveness of lower energy levels for the controls, e.g., by means of optimisation using inequality constraints for the energy instead of equality constraints. This is, however, not in the scope of the current work.

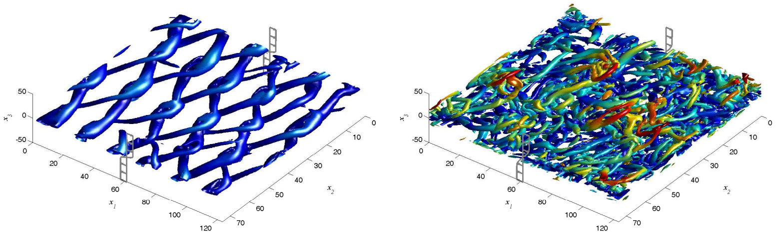
Finally, the coherent structures of optimal solutions are elaborated on in figure 5.13 using the λ_2-criterion (cfr. Jeong and Hussain (1995), and p. 66). In the left-hand column of the figure, the 'optimal' mixing layers are evaluated at $t = 40$, in the right-hand column at $t = 160$. At $t = 40$ all solutions are fairly regular, though some differences may be observed. At $t = 160$ the mixing layers have evolved into a quite complex state, with a very dense packing of small-scale vortices. We now concentrate on $t = 40$ in figure 5.13, corresponding to the initial evolution of the optimal perturbations (λ_2-visualisations at earlier times do not a yield meaningful visualisation of flow structures). For group A, we do not observe meaningful differences between visualisations of flows optimised towards different time horizons. The selected plots in figure 5.13 (a,b) with $T = 120$ are representative for the type of structures observed for other time horizons. For group B, more pronounced differences are visible when the time horizon is changed. The flow structures for group B can be divided into two groups according to their time horizon: $T \leq 80$ and $T \geq 100$. The typical structures for $T \leq 80$ are presented in figure 5.13 (c) using the case $T = 80$. At $t = 40$, structures show a diamond-shape pattern of vorticity in the mixing plane. For $T \geq 100$, and group B typical structures are shown in figure 5.13 (d) (using $T = 120$). The vorticity



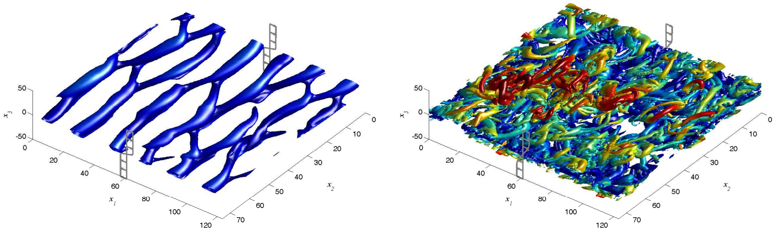
(a) $E_0 = 10^{-5}$, group A, $T = 120$



(b) $E_0 = 10^{-4}$, group A, $T = 120$



(c) $E_0 = 10^{-4}$, group B, $T = 80$



(d) $E_0 = 10^{-4}$, group B, $T = 120$

Figure 5.13: λ_2 visualisation of the velocity-field coloured with the distance from the mixing centre line for parameters optimised towards $\mathscr{J}_{ENS}$; profile $\langle u_1 \rangle$ (—); visualisation of $t = 40$ (left) and $t = 160$ (right)

is now more two-dimensional at $t = 40$, mainly distributed in spanwise vorticity. The flow only shows the diamond-shaped structures later in time.

If the energy distribution of the initial perturbations between the different modes is studied, it is observed that for group B, and $T \geq 100$ about 10% more energy is stored in two-dimensional modes. As a consequence of this (and the more two-dimensional coherent structures) vortex stretching, break-up of vortices, and subsequent dissipation, are 'delayed' to later times compared to the flows in figures 5.13 (a,b,c). As a result the enstrophy rise (e.g. observed in fig. 5.11 (a)) is significantly shifted to larger time horizons. These features are probably related to the better optima found in group B for large time horizons.

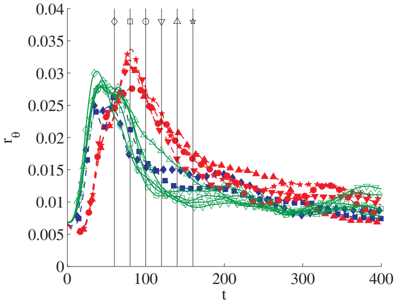
5.4.2 Discussion

The results of the optimisation to $\mathscr{J}_{ENS}$ will now be compared with the optimisation results of $\mathscr{J}_{KE}$ and some differences with respect to the onset of self-similarity are highlighted.

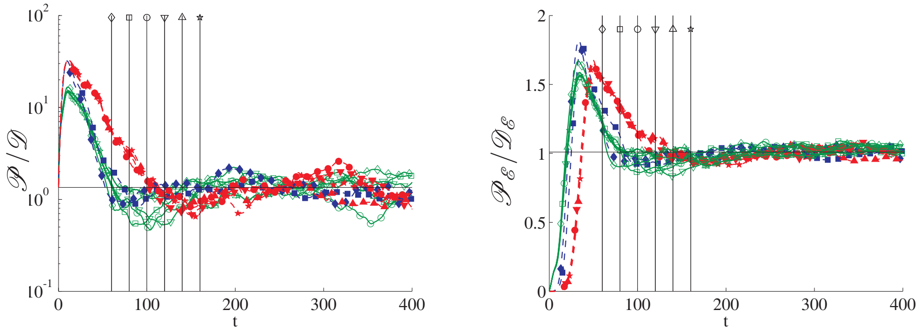
The time evolution of the momentum thickness growth rate, and the ratios $\mathscr{P}/\mathscr{D}$, and $\mathscr{P}_{\mathscr{E}}/\mathscr{D}_{\mathscr{E}}$ (cf. the definitions in (5.5), (5.6), (5.7), and (5.8)) are displayed in figure 5.14 for all optimal solutions from $\mathscr{J}_{ENS}$ with control energy $E_0 = 10^{-4}$. Focusing on the peaks in figures 5.14 (a,b,c), it is observed that the optimal solutions for the dissipation rate roughly fall apart into two sets. The first set consists of all optima from 'group A', and the optima from 'group B' with $T \leq 80$. The second set contains the optima from 'group B' with $T \geq 100$. For the momentum thickness growth rate (fig.5.14 (a)) and the enstrophy growth rate (fig.5.14 (c)) the first set has maxima earlier in time, respectively at $t \approx 50$ and $t \approx 33$, compared to $t \approx 80$ and $t \approx 50$ (see also table 5.1). In figure 5.9 (c) the optima are divided into the aforementioned sets based on the width of the peak of $\mathscr{P}/\mathscr{D}$. This division into two sets corresponds also with the difference in coherent structures (more 3D versus more 2D) observed at the end of section 5.4.1 for all these cases.

Compared to results for optimisation towards maximum dissipated energy ($\mathscr{J}_{KE}$, cf. figures 5.8, 5.9 (a,b)) we find for $\mathscr{J}_{ENS}$ that the peak in r_θ, $\mathscr{P}/\mathscr{D}$, and $\mathscr{P}_{\mathscr{E}}/\mathscr{D}_{\mathscr{E}}$ shifts to later times as set out in table 5.1. Hence, for $\mathscr{J}_{ENS}$, high dissipation rates at the time horizon are realised by the optimal controls via a shift in time of the peak in enstrophy growth rate. These shifts also have a large impact on the onset of self similarity.

Figure 5.14 (a) also shows that all $\mathscr{J}_{ENS}$ optima are still evolving towards self similarity at $t = 400$. Hence, the onset of self-similarity starts significantly later for the flows optimised to the dissipation rate than for the total dissipated energy optimised flow. As an illustration, the normalised profiles of turbulent kinetic



(a) r_θ



(b) $\mathscr{P}/\mathscr{D}$ (c) $\mathscr{P}_\mathscr{E}/\mathscr{D}_\mathscr{E}$

Figure 5.14: (a) Evolution of the momentum thickness growth rate r_θ (b) Evolution of $\mathscr{P}/\mathscr{D}$ in the center of the layer in time. Horizontal line at 1.35 (c) Evolution of the rate of enstrophy production to enstrophy dissipation in time. Horizontal line at 1.01. (a,b,c) Flows optimised towards $\mathscr{J}_{ENS}$ with $T = 60$ ($\Diamond$), $T = 80$ ($\Box$), $T = 100$ ($\circ$), $T = 120$ ($\triangledown$), $T = 140$ ($\triangle$), $T = 160$ ($\star$), A-set (-,empty green symbols), and B-set ($--$, filled symbols, $T \leq 80$ blue, $T \geq 100$ red), $E_0 = 10^{-4}$.

	r_θ	$\mathscr{P}/\mathscr{D}$	$\mathscr{P}_{\mathscr{E}}/\mathscr{D}_{\mathscr{E}}$
$\mathscr{J}_{KE}$	30	8	25
$\mathscr{J}_{ENS}$: group A, group B with $T \leq 80$	50	11	33
$\mathscr{J}_{ENS}$: group B with $T \geq 100$	80	11	50

Table 5.1: Time of the peak in momentum thickness growth rate r_θ, in rate of production to dissipation of turbulent kinetic energy in the centre of the layer $\mathscr{P}/\mathscr{D}$, and in rate of enstrophy production to enstrophy dissipation $\mathscr{P}_{\mathscr{E}}/\mathscr{D}_{\mathscr{E}}$

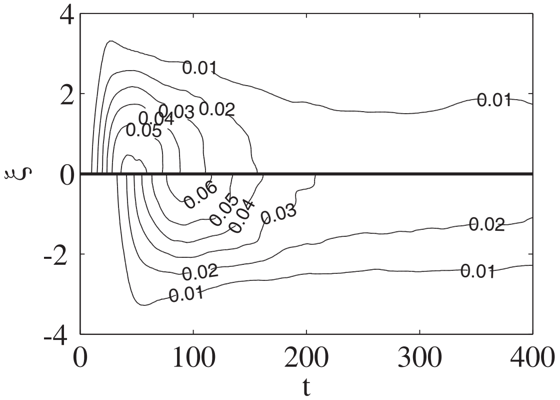


Figure 5.15: Evolution in time of the profile $0.5\langle u'u' + v'v' + w'w' \rangle / \Delta U^2$ as a function of the scaled normal coordinate $\xi = x_3/\theta$. Flows optimised for $T = 160$, $E_0 = 10^{-4}$ to $\mathscr{J}_{KE}$ (upper part figure) and $\mathscr{J}_{ENS}$(lower part figure).

energy as a function of the scaled normal coordinate are depicted in figure 5.15 for $T = 160$ $\mathscr{J}_{KE}$ and $\mathscr{J}_{ENS}$ (group B). The transition to the self-similar state occurs when the contour lines start to 'float around' a horizontal line. The figure illustrates that the profile for $\mathscr{J}_{KE}$ starts to behave self-similar around $t \approx 160-200$, while the profile for $\mathscr{J}_{ENS}$ is still changing at $t = 400$. Thus the dissipation rate may be increased considerably via dedicated optimal controls, but as a result the transition to self-similarity shifts further back in time. This further supports the suggestion that the control of dissipation in shear flows may not be possible once the shear-flow is self similar.

5.5 Conclusion

The ability of flow control to impact upon the dissipation properties of the mixing layer at long time horizons was tested. Optimisation is performed to two different cost functionals to maximise either the total dissipated energy in time interval $[0, T]$, or the dissipation rate at the time horizon T. These two properties evolve respectively linear and constant in time in the self-similar region. Optimisation of the initial perturbations of the mixing layer significantly increased these dissipation properties.

The mixing layer optimised to $\mathscr{J}_{KE}$ was characterised by diamond-shaped structures during the early development followed by a fast onset of self-similarity. The onset occurred at much lower Reynolds numbers than in the mixing layers studied by Rogers and Moser (1994) and Balaras et al. (2001).

The optimisation to $\mathscr{J}_{ENS}$ resulted in a group of optima that is good for the time horizons $T < 100$, and a group of optima that is good for $T \geq 100$. The last group has a higher two-dimensional component than the first group. With regard to the self-similarity: the onset is observed later than for the flow optimised to total dissipation. Maximisation of the rate of dissipation delays the onset of self-similarity.

Different energy levels for the controls were tested. For the total dissipated energy optimisation (using $\mathscr{J}_{KE}$), the effect was straightforward, the higher control energy leads to a higher total dissipation. For the dissipation rate optimisation (using $\mathscr{J}_{ENS}$) the effect was the same for the low time horizons but for the higher time horizons, the dissipation rate was roughly equal for the two energy levels.

Chapter 6

The effect of noise on the optimal perturbations

In the previous chapter, it was shown that careful optimisation of the perturbations on the initial mean flow field can be used to, increase the dissipation properties significantly. It was assumed that the perturbations could be perfectly imposed. The perturbations can be imagined to originate from perturbations on the mean flow field above and below the splitter plate in the spatial mixing layer. It is unlikely that flow control actuation mechanisms on the splitter plate can entirely tailor the perturbations on the mean flow field above and below the plate, for example due to noise coming from upstream. This chapter takes this into account by assuming that the initial field of the temporal mixing layer contains noise:

$$u(\boldsymbol{x}, 0, \boldsymbol{\phi}, \alpha) \quad = \quad \frac{\Delta U}{2} \tanh(x_3)\, \mathbf{e}_1 + \boldsymbol{\phi}(\boldsymbol{x}) + \sqrt{\alpha \frac{2L_3}{\delta_\omega}} \frac{\|\boldsymbol{\phi}\|}{\|\boldsymbol{\varepsilon}\|}\, \boldsymbol{\varepsilon}(\boldsymbol{x}) \tag{6.1}$$

where $\boldsymbol{\phi}$ are the perturbations originating from the control, and $\boldsymbol{\varepsilon}$ the noise. The energy of the noise is $\alpha\ (2L_3)/\delta_\omega$ times the energy of the optimised perturbations. The factor $(2L_3)/\delta_\omega$ (with $\delta_\omega/2$ the reference length) takes into account that the control perturbations are the most important near the mixing plane while the noise is spread out uniformly over the box.

In order to find the optimal control $\boldsymbol{\phi}$ in the presence of noise, in principle optimisation of the perturbations in the presence of uncertainty (here the noise) is required. The cost functional is then considered to be function of the known parameters, $\boldsymbol{\phi}$ and the unknown parameters $\boldsymbol{\varepsilon}$. There are two common approaches in optimisation with uncertainty: stochastic optimisation, and worst-case optimisation (Boyd and Vandenberghe, 2009). In stochastic optimisation the expected performance is optimised (Nocedal and Wright, 2006), to this end $\boldsymbol{\varepsilon}$ is modelled as a random variable with a known distribution (Nocedal and Wright, 2006; Boyd and Vandenberghe, 2009). The cost functional is expressed based on expected values $\boldsymbol{E}_{\boldsymbol{\varepsilon}} \mathscr{J}(\boldsymbol{\phi}, \boldsymbol{\varepsilon})$ where the expectation $\boldsymbol{E}$ is with respect to $\boldsymbol{\varepsilon}$ (Boyd and Vandenberghe, 2009). In worst-case optimisation the cost functional is the maximum of $\mathscr{J}(\boldsymbol{\phi}, \boldsymbol{\varepsilon})$ for $\boldsymbol{\varepsilon} \in E$ with E the set where $\boldsymbol{\varepsilon}$ is known to lie in (Boyd and Vandenberghe, 2009).

Both stochastic optimisation, and worst-case optimisation require the evaluations $\mathscr{J}(\boldsymbol{\phi}, \boldsymbol{\varepsilon})$ for all $\boldsymbol{\varepsilon} \in E$. If the noise is assumed to be representable by 20 realisations, the cost of one cost functional evaluation is twenty flow simulations, which is an increase with a factor 20 compared to chapter 5. Since the number of cost functional evaluations per optimisation in absence of noise in chapter 5 was up to $\mathscr{O}(10^3)$, optimisation in the presence of noise is expected to be computationally very costly. Instead it is opted to investigate to which extent the dissipation properties are affected when white noise is added to the perturbations optimised in absence of noise (presented in chapter 5). The noise level α is gradually increased in subsequent simulations with initial field (eq. (6.1)). The impact on the optimised property and on the start of self-similarity is studied. It is observed that the impact of the presence of noise on the dissipation is small.

In section 5.4 two different groups of optima were found: the optima group A, and group B with $T < 100$, and the optima with a larger 2D component (group B with $T > 100$). The current chapter also addresses the robustness of the division in presence of noise. This chapter was partially presented at DLES8 (Delport et al., 2010b).

6.1 Impact of noise on total dissipated energy

The impact of the noise on the total dissipated energy is studied for optima from section 5.3. In section 5.3.1 it was observed that the early flow evolution is characterised by diamond-shaped coherent structures, which determined to a large extent the total dissipation. The robustness of these structures to the presence of noise is investigated. The optimisation in absence of noise showed that controls that maximise the total dissipated energy fasten up the onset of self-similarity. The current study shows the onset is delayed significantly by the noise.

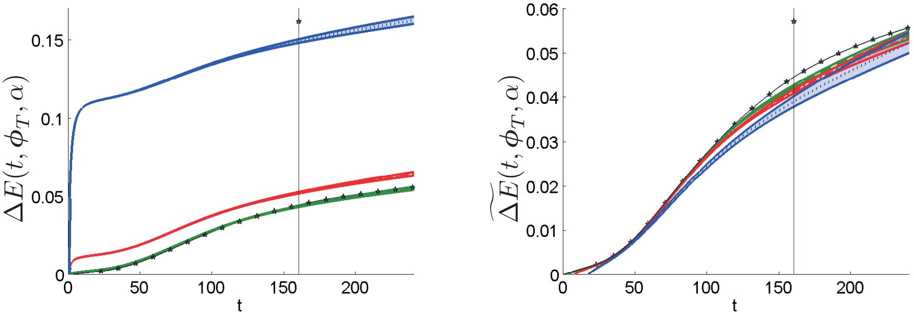
The mixing layer with initial condition (eq. 6.1) is simulated, using $\boldsymbol{\phi}$ optimised in absence of noise. The optimum $\boldsymbol{\phi}_T$, obtained in the section 5.3, that maximised the total dissipation for $T = 160$, $E_0 = 10^{-4}$ is selected for this study as it is also optimal for the lower time horizons. For the noise $\boldsymbol{\varepsilon}$ twenty different realisations are used, corresponding to white noise uniformly distributed over the box. Three levels of noise are considered with $\alpha = 10^{-1}, 1, 10$.

The total dissipation as a function of time is highly dependent on the level of noise, as illustrated in figure 6.1 (a). Initially the dissipation rate of the flow with noise is much larger since the smallest scales that are present in the noise dissipate very fast. This leads to a vertical shift in total dissipation, and complicates the comparison between different noise levels. Therefore the effective dissipation $\widetilde{\Delta E}$ is defined:

$$\begin{aligned}\widetilde{\Delta E}(T, \boldsymbol{\phi}_T, \alpha) &= (1 + E_0)\,\frac{\Delta U^2}{8} - \frac{1}{2}\frac{1}{\Omega}\int_\Omega \boldsymbol{u}(\mathbf{x}, T, \boldsymbol{\phi_T}, \alpha) \cdot \boldsymbol{u}(\mathbf{x}, T, \boldsymbol{\phi_T}, \alpha)\,\mathrm{d}\mathbf{x} \\ &= \Delta E(T, \boldsymbol{\phi}_T, \alpha) - E_0\,\alpha \frac{2L_3}{\delta_\omega}\frac{\Delta U^2}{8} \qquad (6.2)\end{aligned}$$

with $\boldsymbol{u}(\mathbf{x}, T, \boldsymbol{\phi_T}, \alpha)$ the velocity field at the time horizon, and $\Delta E(T, \boldsymbol{\phi}_T, \alpha)$ the total dissipation (analogous to eq. 5.3) up to the time horizon of a mixing layer with initial condition $\boldsymbol{u}(\boldsymbol{x}, 0, \boldsymbol{\phi_T}, \alpha)$ according to equation (6.1). Figure 6.1 (b) shows that the presence of noise decreases the effectiveness of the control to maximise the effective dissipation $\widetilde{\Delta E}$ at $t = 160$ on average with 5%, 7%, and 13% respectively for $\alpha = 0.1$, 1, and 10 due to non linear interactions that alter the optimised flow evolution. The reduction is relatively small taken into account the high level of the noise. Around $t = 30$ the impact on the total dissipated energy is negligible. At this point the coherent structures are diamond shaped. Section 5.3.1 showed that these coherent structures dominate the kinetic energy evolution. Since the effective dissipation is well preserved in presence of noise at $t = 30$, the diamond shaped coherent structures are expected to be robust towards the impact of noise. Figure 6.2 confirms this for $\alpha = 1$, and $\alpha = 10$. For $\alpha = 0.1$ the coherent structures at $t = 30$ (not shown) are the same as in figure 5.6.

Figure 6.3 displays the impact of noise on the onset of self-similarity. The transition to the self-similar state occurs when the contour lines start to float around a horizontal line. The figure illustrates that the profile for $\alpha = 0$ starts to behave self-similar around $t \approx 160 - 200$, while for $\alpha = 10$ the noise delays the onset of self-similarity to $t > 390$. This indicates that a fast onset of self-similarity is connected to an optimal effective dissipation.



(a) Total dissipation (b) Effective total dissipation

Figure 6.1: Evolution in time for optimised $\boldsymbol{\phi}$ without noise (symbols), and with noise (mean$\mp 2\sigma$ band for $\alpha = 0.1$ (green, $--$), $\alpha = 1$ (red, $\cdot -$), $\alpha = 10$ (blue, $\cdots$)). (a) Total dissipation (b) Effective total dissipation

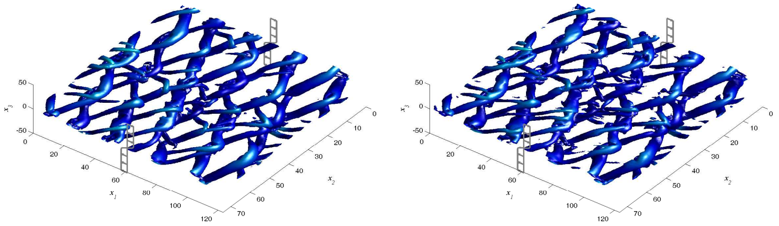


Figure 6.2: λ_2 visualisation of the velocity-field (cfr. Jeong and Hussain (1995), and p. 66) coloured with the distance from the mixing centre line for parameters optimised towards $\mathscr{J}_{KE}$ subject to noise $\alpha = 1$ (left) and $\alpha = 10$ (right); profile $\langle u_1 \rangle$ (—); Visualisation at $t = 30$ for one realisation of noise.

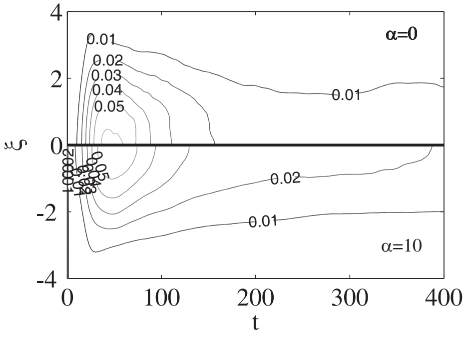


Figure 6.3: Evolution in time of the profile $0.5\langle u'u' + v'v' + w'w' \rangle / \Delta U^2$ in function of the scaled normal coordinate $\xi = x_3/\theta$; in absence of noise (upper part figure), and in presence of noise with level $\alpha = 10$ (lower part figure, averaged over 20 realisations). $\boldsymbol{\phi}$ optimised to $\mathscr{J}_{KE}$ with $T = 160$, $E_0 = 10^{-4}$.

6.2 Impact of noise on dissipation rate

The robustness of the optima for the dissipation rate, found in section 5.4, to noise $\boldsymbol{\varepsilon}$ is also investigated. The mixing layer with initial condition (eq. 6.1) is simulated, using $\boldsymbol{\phi}$ optimised to the enstrophy cost functional in absence of noise. Two optima, corresponding with $\boldsymbol{\phi}_{80}$ and $\boldsymbol{\phi}_{140}$, are selected both from group A and B. For the noise $\boldsymbol{\varepsilon}$, the same levels and realisations as in section 6.1 are used.

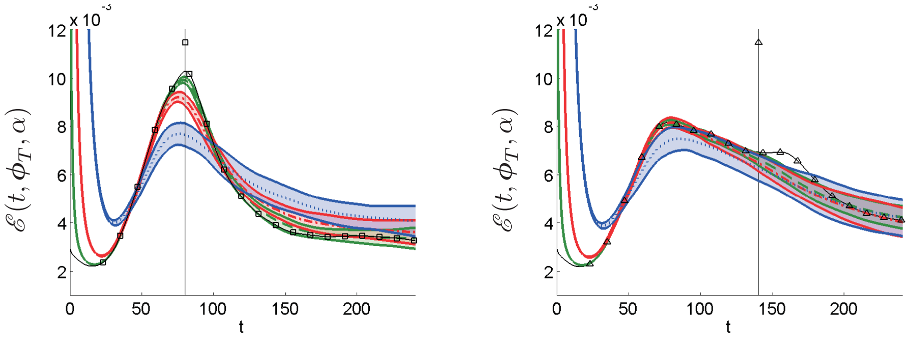
The effect of the noise on the dissipation rate is shown in figure 6.4 based on the enstrophy ($\varepsilon = 2\nu\mathscr{E}$). It is observed that even for noise with much more (factor $10(2\ L_3/\delta_\omega)$) energy in the box than the perturbations $\boldsymbol{\phi}$, the enstrophy evolution with and without noise shows a similar evolution. The mean enstrophy level at the time horizon decreases however with increasing noise level due to non linear interactions between the instabilities triggered by the noise, and the optimised perturbations. These interactions suppress only partially the optimised flow evolution.

Section 5.4 showed that the onset of self-similarity is delayed to maximise the dissipation rate. The presence of noise might trigger the onset of self-similarity faster, reducing the impact of the control at the time horizon. Figures 6.4 shows that this is not the case, and it is appreciated that the optimised parameters are robust enough to delay the onset of self-similarity (the enstrophy is constant in time in the self-similar region). To quantify this more precisely, the momentum thickness growth rate and the turbulent kinetic energy profiles are shown in figures 6.5-6.6 for the $T = 140$ group B optimum subject to noise with $\alpha = 10$. Of the optima depicted in figure 6.4, this optimum delays the onset of self-similarity the most. Figure 6.5 shows the momentum thickness growth rate in the presence of noise. It is noted that the growth rate is still decreasing at $t = 400$, and, as a consequence, has not reached self-similar state yet. This behaviour corresponds with the observations in absence of noise (cfr. section 5.4 and fig. 6.5). Figure 6.6 compares the evolution of the turbulent kinetic energy profiles for the same optimum ($T = 140$ group B) in presence and absence of noise. The onset of self-similarity clearly occurs after $t = 400$ in both cases since the profiles are still changing.

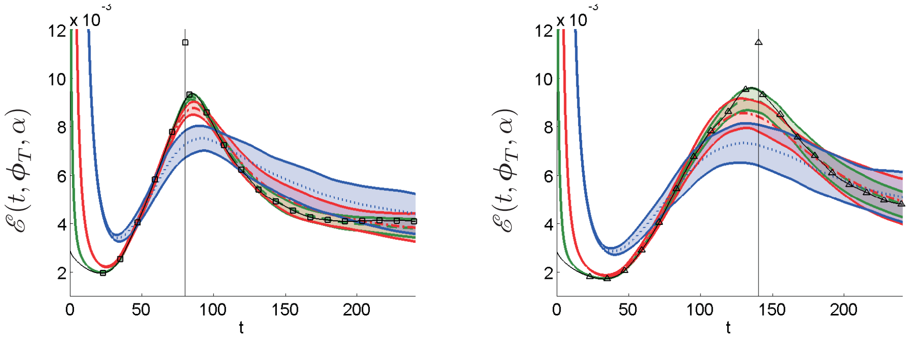
Figure 6.7 illustrates the flow evolution with the parameters subject to noise using the λ_2 criterion (cfr. Jeong and Hussain (1995), and p. 66). For $\alpha = 1$, the impact of the noise on the coherent structures is negligible, and the figure 6.7 (left) is almost equal to figure 5.13 (left). If the noise is increased to $\alpha = 10$ (fig. 6.7 (right)), the number of braid vortices increases but the large vortices change only minimally (fig. 6.7 (left-right)). The features on which the distinction between the group B optima for $T \leq 80$ and $T \geq 100$ is based remained the same. Based on the impact of the noise on the enstrophy evolution, and on the coherent structures we conclude that the optimised states are robust to the impact of noise.

6.3 Conclusion

Noise was added to the optimised perturbations. The subsequent flow evolutions showed a decrease of the effective total dissipation and the dissipation rate (both averaged over the different realisations) that was acceptable even for high levels of noise. The coherent structures of the flow were compared for different noise levels. The characteristic features of the initial transitions remained. We concluded that the optima, both for the total dissipation maximisation and for the dissipation rate maximisation at the time horizon, are robust to the presence of noise. The noise delayed the onset of self-similarity for the total dissipation optimised mixing layer. It is surmised that fast onset of self-similarity is related to maximal effective total dissipation.



(a) group A, $T = 80$ (b) group A, $T = 140$



(c) group B, $T = 80$ (d) group B, $T = 140$

Figure 6.4: Enstrophy evolution in time for optimised $\boldsymbol{\phi}$ without noise (symbols), and with noise (mean$\mp 2\sigma$ band for $\alpha = 0.1$ (green, −−), $\alpha = 1$ (red, ·−), $\alpha = 10$ (blue, ···))

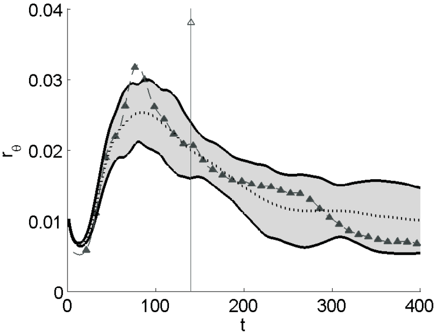


Figure 6.5: Growth rate evolution in time for optimised $\boldsymbol{\phi}$ without noise (symbols), and with noise with level $\alpha = 10$ (mean$\mp 2\sigma$ band $(\cdots)$). $\boldsymbol{\phi}$ optimised to $\mathscr{J}_{ENS}$ with $T = 140$ group B, $E_0 = 10^{-4}$.

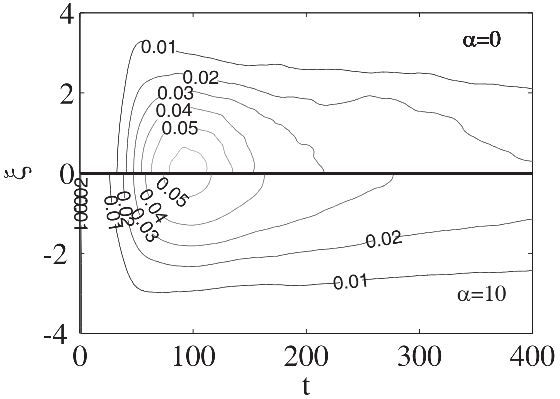
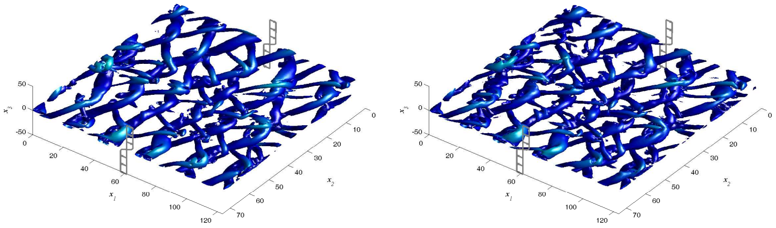
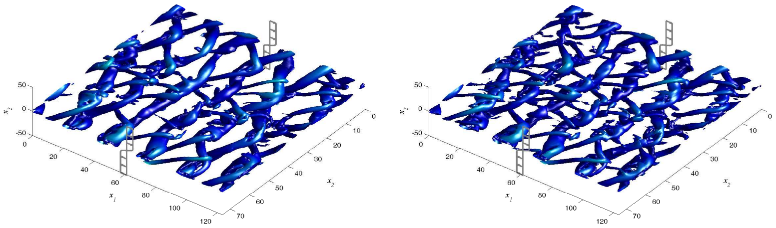


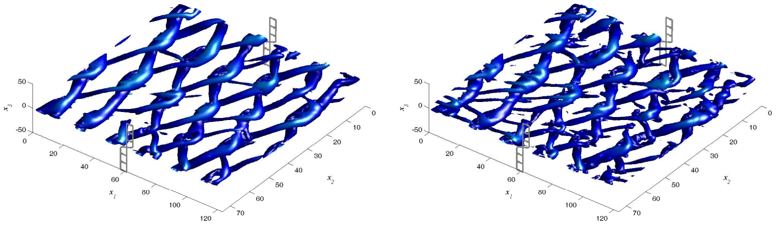
Figure 6.6: Evolution in time of the profile $0.5\langle u'u' + v'v' + w'w' \rangle / \Delta U^2$ in function of the scaled normal coordinate $\xi = x_3/\theta$; in absense of noise (upper part figure), and in presence of noise with level $\alpha = 10$ (lower part figure, averaged over 20 realisations). $\boldsymbol{\phi}$ optimised to $\mathscr{J}_{ENS}$ with $T = 140$ group B, $E_0 = 10^{-4}$.



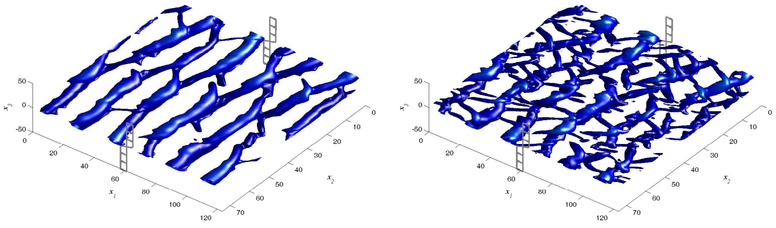
(a) $E_0 = 10^{-4}$, group A, $T = 80$



(b) $E_0 = 10^{-4}$, group A, $T = 140$



(c) $E_0 = 10^{-4}$, group B, $T = 80$



(d) $E_0 = 10^{-4}$, group B, $T = 140$

Figure 6.7: λ_2 visualisation of the velocity-field coloured with the distance from the mixing centre line for parameters optimised towards $\mathscr{J}_{ENS}$ subject to noise $\alpha = 1$ (left) and $\alpha = 10$ (right); profile $\langle u_1 \rangle$ (—). Visualisation at $t = 40$ for one realisation of noise.

Chapter 7

Conclusion and suggestions for future research

7.1 Conclusion

In this work a temporal mixing layer is optimised. The perturbations on the initial mean velocity field are optimised to a variety of properties at the time horizon. The optimisation is subject to a linear and a non-linear constraint, respectively imposing divergence-freeness of the perturbations, and their energy. A range of optimisations is performed with the Reynolds numbers at the time horizon extending as high as $Re_\lambda = 40$.

This work builds on the same techniques as the work of Bewley et al. (2001) and Wei and Freund (2006): gradient-based optimisation with continuous adjoint-based calculation of the gradient. To this end an adjoint simulation is performed whose boundary conditions are imposed by the cost functional. The impact of the cost functional on the choice of the adjoint boundary conditions is detailed for a general flow problem, described by the incompressible Navier-Stokes equations. The derivations show that the formulation of the cost functional is subject to limitations when the adjoint method is used. The adjoint method is applied on the mixing layer optimisation problem and the correct convergence of the gradient as a function of the time step and the grid size is verified.

The PDE-constrained optimisation algorithm (Bewley et al., 2001; Wei and Freund, 2006) is extended to take into account a linear and a non-linear constraint on the control. For the non-linear constraint two methods are compared that comply with the KKT conditions: the gradient-projection method, and the augmented

Lagrangian method. It is found that optimisation with the gradient projection method is the more robust methodology. The augmented Lagrangian method is very sensitive to small gradient inconsistencies which originate from the adjoint-based gradient calculation. As a result, convergence is not always guaranteed.

To further test the optimisation method, and to investigate the impact of the cost functional, optimisation is performed to five different cost functionals at two different time horizons $T = 20$ and 40. The cost functionals are based on the momentum thickness, turbulent kinetic energy, mean-flow kinetic energy, kinetic energy, or enstrophy. The cost functionals that optimise the momentum thickness, the mean-flow kinetic energy, or the turbulent kinetic energy are shown to lead to the same optima. They resulted in large two-dimensional vortex structures with maximum impact on mean momentum, but low dissipation. These structures are similar to the structures that develop starting from a combination of 2D LST modes, but it is shown that the mixing layer with the optimised modes grows about 40% faster (based on the turbulent kinetic energy) since the optimisation takes into account the non-linear interactions between the different modes. The other two cost-functionals, based on the total kinetic energy and the enstrophy, resulted in optimal solutions that have complex three-dimensional vortex structures at different scales.

During the optimisation to minimise the total kinetic energy, it was observed that the landscape in which the optimisation is performed is strongly non convex, and that the optimum for $T = 40$ is a local optimum. This is concluded as the parameters optimised for $T = 20$ lead to a lower total kinetic energy at $t = 40$ than the parameters optimised for $T = 40$. To further investigate this, the optimal perturbations obtained for $T = 20$ are used as a starting point for optimisation with $T = 40$, and it is found that the optimisation with time window $T = 40$ now converged to a set of perturbations which are close to those found for $T = 20$. Based on this experience, a continuation procedure is proposed that starts the optimisation to a time horizon T_2 from the optimum for the preceding time horizon T_1.

The last part of this work shows that the mixing-layer properties can be optimised up to long time horizons. The kinetic energy, and the enstrophy cost functionals are used to respectively maximise the total dissipated energy in an interval $[0, T]$, and the rate of dissipation at the time horizon T. Both dissipation properties are significantly increased compared to the unoptimised flows for all time horizons used. The Taylor-Reynolds number at the time horizons reaches $Re_\lambda \approx 40$, indicating that the mixing layer is optimised at time horizons in the fully turbulent regime.

It is also studied if the optimisation of the perturbations impacts the self-similar properties of the mixing layer. There are no indications that the 30 different optimisation cases force the mixing layer to different self-similar states. This does not exclude the existence of different self-similar states, but it does not confirm it either.

However, a shift in the moment of the onset to self-similarity is observed. When the mixing-layer evolution is evaluated using the optimised initial perturbations, it is observed that optimisation of the dissipated energy fastens up the evolution to a self-similar state. The onset of self-similarity occurs at much lower Reynolds numbers ($Re_\theta \approx 350$) than in mixing-layer studies by Rogers and Moser (1994) and Balaras et al. (2001). For optimisation of the dissipation rate, on the other hand, it is found that self-similarity starts much later.

Finally, the robustness of the optima is studied by adding noise to the optimised perturbations. The subsequent flow evolutions show a decrease of the effective total dissipation and the dissipation rate (both averaged over the different realisations of noise) that is acceptable even for high levels of noise. It is concluded that the optima, of both cost functionals, are robust in presence of noise. The noise delays the onset of self-similarity of the mixing layer optimised for the total dissipated energy.

7.2 Suggestions for future research

There are some interesting questions that remain, and new questions are formulated. Most of them are computationally so challenging that they require improvement of the optimisation algorithm, and computational speed-up. Ideas to enhance the optimisation algorithm are discussed in section 7.2.1, afterwards interesting optimisation cases are detailed in section 7.2.2.

7.2.1 Optimisation algorithm

The current research faced a high computational cost. The calculation time for the optimisation was for some time horizons more than a month on 32 CPU's, and the need for scratch storage space to store the forward field, which is required during the backward time integration of the adjoint system, was up to 660 GB (for $T = 160$ on a $128 \times 128 \times 256$ mesh like in chapter 5). Therefore future research may be directed towards reduction of the computational time and storage need.

The *computational time* is determined by the calculation cost per flow simulation, and the number of cost functional evaluations. The computational cost per flow simulation can be reduced by using LES. This implies the formulation of an adjoint subgrid scale model. To reduce the number of cost functional evaluations, it might be worthwhile to compare the performance of limited memory BFGS with the Polak-Ribière method for the current optimisation problem. Another track is to reduce the parameter space as the number of cost functional evaluations increases with the number of parameters. Two possibilities to reduce the number of optimisation parameters are to define the eigenfunctions of mode $(\alpha, -\beta)$ as opposite/equal to (α, β) or, for the total dissipated energy optimisation, to optimise only the profiles of the modes that have a large amount of energy. Reducing the parameter space also has the advantage of decreasing the number of local optima. If optimisation is desired in the large parameter-space, it is interesting to test a multi-grid approach for the optimisation. First the initial perturbations are optimised on a coarse grid. The optimal perturbations are subsequently mapped to a finer grid and the optimisation is continued. The result is again mapped to a finer grid, etc. Finally the optimum on the fine grid is obtained. Since the optimisation on the finest grid starts from a very good guess for the optimal parameters, it will probably converge fast, and the total computational time might be reduced.

The *storage need* is determined by the fields that are stored during the forward calculation to be used during the backward time integration of the adjoint system. Currently the velocity at each grid point and each time step is stored. During the backward time integration the fields are approximated piecewise constant in time (cfr. section 4.1). By using higher order interpolation, it might be satisfactory to store the velocity field only every n time steps, and to use interpolation for the intermediate fields. This would reduce the disk-storage requirements by a factor n. Wei and Freund (2006), and Bewley et al. (2001) used a respectively a factor 2 and 5 in combination with linear interpolation. Another interesting technique to reduce the storage requirements is to store the velocity field only at every other point and to interpolate in space during the backward calculation. Wei and Freund (2006) applied this technique and used third-order for the interpolation. Analysing the error on the gradient for different interpolation methods would be interesting. An alternative is to use checkpointing, which was discussed in 3.2.2. Hinze and Sternberg (2005) showed that a memory reduction of two orders of magnitude can be achieved at the cost of a factor 2 à 3 increase in run-time.

7.2.2 Interesting optimisation cases

By extending the current work, some interesting questions can be answered. Four topics are listed here. The list starts with the research that requires almost no code development, and ends with the topic that requires the most development.

The optimisation of the dissipation rate delays the onset of self-similarity, such that the dissipation rate can be maximised. If the self-similarity was not delayed, the dissipation rate would be constant and determined by the self-similar state. In section 5.1.1 the ongoing discussion about the existence of several self-similar states was mentioned. In this work there were no indications that there are several self-similar states (cfr. chapter 5). If there is only one self-similar state, it is not possible to influence the dissipation rate after the onset of self-similarity. It would be interesting to investigate up to which time horizon the dissipation rate can be optimised, as it is unlikely that the self-similar region can be delayed forever, especially in presence of noise. This requires optimisation in larger boxes, which is computationally more demanding.

With regard to the energy constraint, it would be interesting to test the (in)effectiveness of lower energy levels of the control by imposing the energy constraint with an inequality instead of an equality constraint. For the lower time horizons, I expect the same results. For the higher time horizons, on the other hand, the optimisation of the dissipation rate showed that controls with a lower energy level might lead to a higher dissipation rate.

Since a mixing layer depicts a basic phenomenon that occurs in many mixing processes, it would also be interesting to perform optimisation to mixing. The diamond-shaped coherent structures observed for the mixing layer optimised to maximal total dissipated energy are similar to the structures in Nygaard and Glezer (1990). Collis et al. (1994) showed that these diamond-shaped structures are good for mixing. Using optimisations to a cost functional measuring the distribution of a passive scalar, it can be investigated if these structures are the most optimal for mixing.

The current code is very efficient thanks to the pseudo-spectral discretisation and the corresponding convergence. The disadvantage is the obligation to use periodic boundary conditions in two directions. With a more versatile 3D finite volume code, flow control on the splitter plate can be explicitly optimised, leading to optimisation results that can be translated into actuation mechanisms for an experiment set-up. Such a versatile code (Lardat and Leschziner, 1998) is currently extended to enable optimisation towards drag reduction for a backward facing step and would also allow optimisation of a spatial mixing layer. The expected time per simulation is however much larger and it will be necessary to lower the maximum number of conjugate gradient iterations to limit the number of flow simulations.

To conclude, there are many interesting research tracks starting from this research, and I sincerely hope this work is a sound basis for continued research on optimisation of open-loop control for mixing layers.

Appendix A

Derivation of adjoint spatial boundary conditions

In section 2.4.1 the Gateaux differential was written as (eq. (2.38)):

$$\begin{aligned}\delta\mathscr{J}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) &= (\boldsymbol{q}^*, f') \\ &+\delta\mathscr{J}_{\partial\Omega}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) - BT_{\partial\Omega} \\ &+\delta\mathscr{J}_T(\boldsymbol{\phi};\delta\boldsymbol{\phi}) - BT_T \\ &-BT_0. \end{aligned} \tag{A.1}$$

The right-hand side of (A.1) is divided into parts that integrate over the same boundary, i.e. $t = 0$, $t = T$, or $\partial\Omega$. To obtain the efficiency increase promised in section 2.4 for the adjoint method, each of these parts should not require the calculation of $\delta\boldsymbol{q}$ through a flow simulation. This requirement for the part integrating over the boundary $\partial\Omega_j$ (normal to direction n_j) is:

$$\begin{aligned} &\delta\mathscr{J}_{\partial\Omega_j}(\phi;\delta\boldsymbol{\phi}) - BT_{\partial\Omega_j} \\ = \;& \delta\mathscr{J}_{\partial\Omega_j}(\phi;\delta\boldsymbol{\phi}) - \int_0^T \int_{\partial\Omega_j} \left((u_i'u_j + u_iu_j')u_i^* + p'u_j^* + u_j'p^*\right) n_j \mathrm{d}\mathbf{x}\mathrm{d}t \\ & - \int_0^T \int_{\partial\Omega_j} \frac{1}{Re}\left(-\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right)u_i^* + u_i'\left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i}\right)\right) n_j \mathrm{d}\mathbf{x}\mathrm{d}t \\ = \;& (f(\boldsymbol{q},\boldsymbol{q}^*), g(\delta\boldsymbol{q}))_{\partial\Omega_j} \end{aligned} \tag{A.2}$$

with f and g denoting analytical functions.

Independent of the (linearised) Navier–Stokes boundary conditions, equation (A.2) limits the Gateaux differential of the cost functional $\delta\mathscr{J}_{\partial\Omega_j}(\phi;\delta\boldsymbol{\phi})$ to (cfr. eq. (2.43))

$$\delta\mathscr{J}_{\partial\Omega_j}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) \quad = \quad \int_0^T \int_{\partial\Omega_j} a_i u_i' + b\, p' + c_i \left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) n_j \,\mathrm{d}\mathbf{x}\mathrm{d}t, \qquad \text{(A.3)}$$

as the $\delta\boldsymbol{q}$ in this differential are the only one represented in $BT_{\partial\Omega_j}$. According to equations (A.2-A.3) the requirement is:

$$\begin{aligned}
&\int_0^T \int_{\partial\Omega_j} a_i u_i' + b\, p' + c_i \left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) n_j \,\mathrm{d}\mathbf{x}\mathrm{d}t \\
&-\int_0^T \int_{\partial\Omega_j} \left((u_i' u_j + u_i u_j')u_i^* + p' u_j^* + u_j' p^*\right) n_j \mathrm{d}\mathbf{x}\mathrm{d}t \\
&-\int_0^T \int_{\partial\Omega_j} \frac{1}{Re}\left(-\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) u_i^* + u_i' \left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i}\right)\right) n_j \mathrm{d}\mathbf{x}\mathrm{d}t \\
= \quad &(f(\boldsymbol{q},\boldsymbol{q}^*), g(\delta\boldsymbol{q}))_{\partial\Omega_j} \qquad \text{(A.4)}
\end{aligned}$$

After rearranging the terms this corresponds with:

$$\begin{aligned}
&\int_0^T \int_{\partial\Omega_j} u_i' \left(a_i - u_j u_i^* n_j - u_j u_j^* n_i - p^* n_i - \frac{1}{Re}\left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i}\right) n_j\right) \mathrm{d}\mathbf{x}\mathrm{d}t \\
&+\int_0^T \int_{\partial\Omega_j} p'(b - u_j^* n_j) \,\mathrm{d}\mathbf{x}\mathrm{d}t \\
&+\int_0^T \int_{\partial\Omega_j} \left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right)\left(c_i + \frac{1}{Re} u_i^*\right) n_j \,\mathrm{d}\mathbf{x}\mathrm{d}t \\
= \quad &(f(\boldsymbol{q},\boldsymbol{q}^*), g(\delta\boldsymbol{q}))_{\partial\Omega_j} \qquad \text{(A.5)}
\end{aligned}$$

Depending on the boundary conditions of the forward problem some properties of $\delta\boldsymbol{q}$ may depend directly on $\boldsymbol{\phi}$, and others will require a flow simulation. The coefficients of the last category ($\delta\boldsymbol{q} \neq g(\delta\boldsymbol{\phi})$) have to be set to zero by selection of the adjoint boundary conditions. In the remainder of the appendix, the adjoint boundary conditions are derived for a symmetry boundary, periodic boundary, and a boundary with velocity control (in the forward domain).

A.1 Symmetry boundary condition

At the $x_3 = \mp L_3/2$ boundary of the domain, with normal $\mathbf{n} = (0, 0, \mp 1)$, symmetry boundary conditions are used:

$$u_3 = 0, \qquad \frac{\partial u_1}{\partial x_3} = 0, \qquad \frac{\partial u_2}{\partial x_3} = 0, \tag{A.6}$$

$$u_3' = 0, \qquad \frac{\partial u_1'}{\partial x_3} = 0, \qquad \frac{\partial u_2'}{\partial x_3} = 0, \tag{A.7}$$

Using the boundary conditions for u_3, and u_3' (A.6-A.7), it is also clear that:

$$\frac{\partial u_3}{\partial x_1} = \frac{\partial u_3}{\partial x_2} = 0, \tag{A.8}$$

$$\frac{\partial u_3'}{\partial x_1} = \frac{\partial u_3'}{\partial x_2} = 0, \tag{A.9}$$

Taking (A.6-A.9) into account, requirement (A.5) imposes for the symmetry boundary Ω_s at $x_3 = L_3/2$ with normal $(0, 0, 1)$:

$$\begin{aligned}
& \int_0^T \int_{\partial\Omega_s} u_1' \left(a_1 - \frac{1}{Re} \left(\frac{\partial u_1^*}{\partial x_3} + \frac{\partial u_3^*}{\partial x_1} \right) \right) \mathrm{d}\mathbf{x}\mathrm{d}t \\
& \int_0^T \int_{\partial\Omega_s} u_2' \left(a_2 - \frac{1}{Re} \left(\frac{\partial u_2^*}{\partial x_3} + \frac{\partial u_3^*}{\partial x_2} \right) \right) \mathrm{d}\mathbf{x}\mathrm{d}t \\
& + \int_0^T \int_{\partial\Omega_s} p' (b - u_3^*) \, \mathrm{d}\mathbf{x}\mathrm{d}t \\
& + \int_0^T \int_{\partial\Omega_s} 2 \, \frac{\partial u_3'}{\partial x_3} \left(c_3 + \frac{1}{Re} u_3^* \right) \mathrm{d}\mathbf{x}\mathrm{d}t \\
= \quad & (f(\boldsymbol{q}, \boldsymbol{q}^*), g(\delta \boldsymbol{q}))_{\partial\Omega_s}
\end{aligned} \tag{A.10}$$

To fulfil the above equation, the boundary conditions of the adjoint have to be:

$$u_3^* = b = -c_3\ Re \tag{A.11}$$

$$\frac{\partial u_1^*}{\partial x_3} = \left(a_1 + \frac{\partial c_3}{\partial x_1}\right)\ Re \tag{A.12}$$

$$\frac{\partial u_2^*}{\partial x_3} = \left(a_2 + \frac{\partial c_3}{\partial x_2}\right)\ Re \tag{A.13}$$

If the adjoint boundary conditions are selected in this way, the Gateaux differential of the cost functional at the upper boundary has the format:

$$\delta \mathscr{J}_{\partial\Omega}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) = \int_{T_1}^{T_2}\int_{\partial\Omega} a_i u_i' - c_3\ Re\ p' + c_i\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) n_j\ \mathrm{d}\mathbf{x}\mathrm{d}t \tag{A.14}$$

For the temporal mixing-layer optimisation, the cost functional does not depend on the velocity, or pressure at the symmetry boundary. As a consequence, a_i, b, and c_i are zero, and equations (A.11-A.13) impose symmetry boundary conditions on the adjoint problem at the $x_3 = L_3/2$ boundary of the domain. For the lower boundary also symmetry boundary conditions are found in an analogous way.

A.2 Periodic boundary condition

When the Navier-Stokes problem has periodic boundary conditions, the velocity and pressure on the boundary with periodic relationship, $\partial\Omega_{p1}$ and $\partial\Omega_{p2}$, are equal and the normals $\mathbf{n}$ are opposite. In that case, the cost functional depending on the periodic boundaries can limit its dependence to the properties at one of the boundaries, e.g. $\partial\Omega_{p1}$. The left-hand side of (A.5) for boundary $\partial\Omega_{p2}$ can then be written as an integral over the boundary $\partial\Omega_{p1}$:

$$\begin{aligned}
& \int_0^T \int_{\partial\Omega_{p2}} u_i' \left(-u_j u_i^* n_j - u_j u_j^* n_i - p^* n_i - \frac{1}{Re} \left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right) n_j \right) \,\mathrm{d}\mathbf{x}\mathrm{d}t \\
& + \int_0^T \int_{\partial\Omega_{p2}} p' (-u_j^* n_j) \,\mathrm{d}\mathbf{x}\mathrm{d}t \\
& + \int_0^T \int_{\partial\Omega_{p2}} \left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i} \right) \left(\frac{1}{Re} u_i^* \right) n_j \,\mathrm{d}\mathbf{x}\mathrm{d}t \\
= \; & \int_0^T \int_{\partial\Omega_{p1}} u_i' \left(u_j u_i^*(\partial\Omega_{p2}) n_j + u_j u_j^*(\partial\Omega_{p2}) n_i + p^*(\partial\Omega_{p2}) n_i \right) \,\mathrm{d}\mathbf{x}\mathrm{d}t \\
& + \int_0^T \int_{\partial\Omega_{p1}} u_i' \left(\frac{1}{Re} \left(\frac{\partial u_i^*}{\partial x_j}(\partial\Omega_{p2}) + \frac{\partial u_j^*}{\partial x_i}(\partial\Omega_{p2}) \right) n_j \right) \,\mathrm{d}\mathbf{x}\mathrm{d}t \\
& + \int_0^T \int_{\partial\Omega_{p1}} p' (u_j^*(\partial\Omega_{p2}) n_j) \,\mathrm{d}\mathbf{x}\mathrm{d}t \\
& + \int_0^T \int_{\partial\Omega_{p1}} \left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i} \right) \left(\frac{1}{Re} u_i^*(\partial\Omega_{p2}) \right) n_j \,\mathrm{d}\mathbf{x}\mathrm{d}t \qquad \text{(A.15)}
\end{aligned}$$

where $u_i^*(\partial\Omega_{p2})$ and $p^*(\partial\Omega_{p2})$ are respectively the adjoint velocity and adjoint pressure on the $\partial\Omega_{p2}$ boundary.

The $\boldsymbol{q}^*$ terms in (A.5) for $\partial\Omega_j = \partial\Omega_{p1} + \partial\Omega_{p2}$ will be determined such that it contains no longer $\delta\boldsymbol{q}$ terms. In the derivation only the difference between $\boldsymbol{q}^*$ at the two related boundaries ($\partial\Omega_{p1}$ and $\partial\Omega_{p2}$) is important. This difference is abbreviated as $\Delta a = a(\partial\Omega_{p1}) - a(\partial\Omega_{p2})$. Following from equations (A.5,A.15) :

$$\begin{aligned}
& \int_0^T \int_{\partial\Omega_{p1}} u_i' \left(a_i - u_j \Delta u_i^* n_j - u_j \Delta u_j^* n_i - \Delta p^* n_i \right) \, \mathrm{d}\mathbf{x}\mathrm{d}t \\
& \int_0^T \int_{\partial\Omega_{p1}} u_i' \left(-\frac{1}{Re} \Delta \left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right) n_j \right) \, \mathrm{d}\mathbf{x}\mathrm{d}t \\
& + \int_0^T \int_{\partial\Omega_{p1}} p' (b - \Delta u_j^* n_j) \, \mathrm{d}\mathbf{x}\mathrm{d}t \\
& + \int_0^T \int_{\partial\Omega_{p1}} \left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i} \right) \left(c_i + \frac{1}{Re} \Delta u_i^* \right) n_j \, \mathrm{d}\mathbf{x}\mathrm{d}t \\
= \quad & (f(\boldsymbol{u}, \Delta \boldsymbol{u}^*), g(\delta \boldsymbol{\phi}))_{\partial\Omega_{p1}} \qquad \text{(A.16)}
\end{aligned}$$

The coefficients of u_i' and p' in (A.16) have to be zero to fulfil the requirement, since u_i' and p' are not an explicit function of $\boldsymbol{\phi}$ ($\delta \boldsymbol{q} \neq g(\delta\phi)$). Therefore the following set of equations should apply:

$$a_i - u_j(\Delta u_i^* n_j + \Delta u_j^* n_i) - \Delta p^* n_i - \frac{1}{Re} \Delta \left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i} \right) n_j = 0 \qquad \text{(A.17)}$$

$$(u_i^*(\partial\Omega_{p1}) - u_i^*(\partial\Omega_{p2})) n_i = b = -Re \cdot c_i \, n_i \qquad \text{(A.18)}$$

$$u_i^*(\partial\Omega_{p1}) - u_i^*(\partial\Omega_{p2}) = -Re \cdot c_i \qquad \text{(A.19)}$$

For general a_i, b, and c_i, this set of equations (A.17-A.19) overdetermines the boundary conditions, since only three degrees of freedom can be fixed. However for $a_i = b = c_i = 0$ on the periodic boundary, it can be satisfied by selecting periodic boundary conditions for the adjoint problem.

For the temporal mixing layer, the boundary conditions in streamwise and spanwise direction are therefore periodic in the forward and the backward simulation.

A.3 Velocity control at the boundary

The two boundary conditions discussed so far are used in the temporal mixing layer case studied. In this section, a boundary condition is discussed that is often used to represent flow control. It is a boundary with velocity control:

$$u_i = \boldsymbol{\phi}_i, \tag{A.20}$$

$$u'_i = \delta\boldsymbol{\phi}_i \tag{A.21}$$

For $\boldsymbol{\phi}_i = 0$, this boundary condition corresponds to a wall boundary condition (no control). The corresponding adjoint boundary conditions are discussed at the end of this section. First the adjoint boundary conditions are derived for any $\boldsymbol{\phi}_i$ based on requirement (A.5). In case of velocity control on the boundary $\partial\Omega_c$ (A.5) simplifies to:

$$\begin{aligned} & \int_0^T \int_{\partial\Omega_c} \delta\boldsymbol{\phi}_i \left(a_i - \boldsymbol{\phi}_j u_i^* n_j - \boldsymbol{\phi}_j u_j^* n_i - p^* n_i - \frac{1}{Re}\left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i}\right) n_j \right) \mathrm{d}\mathbf{x}\mathrm{d}t \\ & + \int_0^T \int_{\partial\Omega_c} p'(b - u_j^* n_j)\, \mathrm{d}\mathbf{x}\mathrm{d}t \\ & + \int_0^T \int_{\partial\Omega_c} \left(\frac{\partial u'_i}{\partial x_j} + \frac{\partial u'_j}{\partial x_i}\right)\left(c_i + \frac{1}{Re} u_i^*\right) n_j\, \mathrm{d}\mathbf{x}\mathrm{d}t \\ = \quad & (f(\boldsymbol{q}, \boldsymbol{q}^*), g(\delta\boldsymbol{q}))_{\partial\Omega_c} \end{aligned} \tag{A.22}$$

p' is not an explicit function of $\delta\boldsymbol{\phi}$, and it can only be determined through a flow simulation depending on $\delta\boldsymbol{\phi}$. To eliminate the integral on the left-hand side of (A.22) with p' involved, the coefficient of p' has to be made zero, leading to (A.23). The velocity derivatives in the wall-normal direction are also unknown without a flow simulation, therefore (A.24) needs to be imposed.

$$u_j^* n_j = b \tag{A.23}$$

$$u_i^* = -c_i\, Re \tag{A.24}$$

Equations (A.23-A.24) impose $-c_i n_i Re = b$, and as such restrict the formulation of the Gateaux differential of the cost functional on the spatial boundary $\partial\Omega_c$ (A.3) to:

$$\delta\mathscr{J}_{\partial\Omega_c}(\phi; \delta\boldsymbol{\phi}) \quad = \quad \int_0^T \int_{\partial\Omega_c} a_i u'_i - Re\ c_i n_i\ p' + c_i \left(\frac{\partial u'_i}{\partial x_j} + \frac{\partial u'_j}{\partial x_i}\right) n_j\, \mathrm{d}\mathbf{x}\mathrm{d}t \tag{A.25}$$

The Gateaux differential of the total cost functional (A.1) is then (using (2.35), (2.41), and (A.22)):

$$\begin{aligned} & \delta\mathscr{J}(\boldsymbol{\phi};\delta\boldsymbol{\phi}) \qquad\qquad (A.26)\\ = \quad & (\boldsymbol{q}^*, f') + (\boldsymbol{u}^*, \delta\boldsymbol{u})_0 \\ + \quad & \int_0^T \int_{\partial\Omega_c} \delta\boldsymbol{\phi}_i \left(a_i + Re\ \boldsymbol{\phi}_j (n_j c_i + n_i c_j) - p^* n_i - \frac{1}{Re}\left(\frac{\partial u_i^*}{\partial x_j} + \frac{\partial u_j^*}{\partial x_i}\right) n_j \right) \mathrm{d}\mathbf{x}\mathrm{d}t. \end{aligned}$$

which does not require a Navier–Stokes simulation to define unknown $\delta\boldsymbol{q}$.

In case the spatial boundary is a wall ($u_i = 0$), then $\delta\boldsymbol{\phi}_i = 0$, so there is no contribution of the boundary to the Gateaux differential (cfr. (A.26)). The adjoint boundary conditions correspond with (A.23-A.24). Note that if the cost functional measures the flow on a certain part of the wall, this part of the wall has a Dirichlet boundary condition during the adjoint simulation.

Bibliography

P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds.* Princeton University Press, Princeton, NJ, US, 2008.

W. K. Anderson and V. Venkatakrishnan. Aerodynamic design optimization on unstructured grids with a continuous adjoint formulation. *Computers & Fluids*, 28(4-5):443–480, 1999.

E. Balaras, U. Piomelli, and J. M. Wallace. Self-similar states in turbulent mixing layers. *Journal of Fluid Mechanics*, 446:1–24, 2001.

R. G. Batt. Turbulent mixing of passive and chemically reacting species in a low-speed shear-layer. *Journal of Fluid Mechanics*, 82(AUG19):53–95, 1977.

J. H. Bell and R. D. Mehta. Development of a 2-stream mixing layer from tripped and untripped boundary-layers. *AIAA Journal*, 28(12):2034–2042, 1990.

L. P. Bernal and A. Roshko. Streamwise vortex structure in plane mixing layers. *Journal of Fluid Mechanics*, 170:499–525, 1986.

T. R. Bewley, P. Moin, and R. Temam. DNS-based predictive control of turbulence: an optimal benchmark for feedback algorithms. *Journal of Fluid Mechanics*, 447: 179–225, 2001.

S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambridge University Press, New York, 7th edition, 2009.

R. P. Brent. *Algorithms for minimization without derivatives.* Series in automatic computation. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1973.

J. Brezillon and N. R. Gauger. 2D and 3D aerodynamic shape optimisation using the adjoint approach. *Aerospace Science and Technology*, 8(8):715–727, 2004.

G. L. Brown and A. Roshko. On density effects and large structure in turbulence mixing layer. *Journal of Fluid Mechanics*, 64:775, 1974.

J. Burkardt, M. Gunzburger, and J. Peterson. Insensitive functionals, inconsistent gradients, spurious minima, and regularized functionals in flow optimization problems. *International Journal of Computational Fluid Dynamics*, 16(3):171–185, 2002.

C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, editors. *Spectral methods, fundamentals in single domains.* Scientific computation. Springer, Berlin, 2006.

L. N. Cattafesta, Q. Song, D. R. Williams, C. W. Rowley, and F. S. Alvi. Active control of flow-induced cavity oscillations. *Progress in Aerospace Sciences*, 44 (7-8):479–502, 2008.

H. Choi, W. P. Jeon, and J. Kim. Control of flow over a bluff body. *Annual Review of Fluid Mechanics*, 40:113–139, 2008.

E. K. P. Chong and S. H. Zak. *An introduction to optimization.* Wiley-interscience series in discrete mathematics and optimization. Wiley, New York, 1996.

S.S. Collis, S.K. Lele, R.D. Moser, and M.M. Rogers. The evolution of a plane mixing layer with spanwise nonuniform forcing. *Physics of Fluids*, 6(1):381–395, 1994.

P. Comte, M. Lesieur, and E. Lamballais. Large-scale and small-scale stirring of vorticity and a passive scalar in a 3-D temporal mixing layer. *Physics of Fluids a-Fluid Dynamics*, 4(12):2761–2778, 1992.

A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-region methods.* MPS-SIAM series on optimization; 1. SIAM, Philadelphia (PA), 2000.

S. Delport, M. Baelmans, and J. Meyers. Optimization of turbulent mixing restricted by linear and nonlinear constraints. In *ERCOFTAC WORKSHOP Direct and Large-Eddy Simulation 7 (DLES7)*, Trieste - Italy, 2008.

S. Delport, M. Baelmans, and J. Meyers. Constrained optimization of turbulent mixing-layer evolution. *Journal of Turbulence*, 10(18):26, 2009.

S. Delport, M. Baelmans, and J. Meyers. Optimization of long-term mixing in a turbulent mixing layer. In *5th AIAA Flow Control Conference*, volume AIAA 2010-4422, Chicago - IL, US, 2010a.

S. Delport, M. Baelmans, and J. Meyers. The effect of noise on optimal perturbations for turbulent mixing. In *ERCOFTAC WORKSHOP Direct and Large-Eddy Simulation 8 (DLES8)*, Eindhoven - The Netherlands, 2010b.

P. E. Dimotakis and G. L. Brown. Mixing layer at high Reynolds-number - large-structure dynamics and entrainment. *Journal of Fluid Mechanics*, 78(DEC7): 535–560, 1976.

B. Epstein and S. Peigin. Constrained aerodynamic optimization of three-dimensional wings driven by Navier-Stokes computations. *AIAA Journal*, 43 (9):1946–1957, 2005.

M. Fathali, J. Meyers, G. Rubio, S. Smirnov, and M. Baelmans. Sensitivity analysis of initial condition parameters on the transitional temporal turbulent mixing layer. *Journal of Turbulence*, 9(N12), 2008.

J. H. Ferziger and M. Peric. *Computational methods for fluid dynamics*. Springer, Heidelberg, Germany, 3rd edition, 2002.

R. Fletcher. *Practical methods of optimization*. Wiley, Chichester, 2nd edition, 2000.

K. Fukagata, S. Kern, P. Chatelain, P. Koumoutsakos, and N. Kasagi. Evolutionary optimization of an anisotropic compliant surface for turbulent friction drag reduction. *Journal of Turbulence*, 9(35):1–17, 2008.

W. K. George. The self-preservation of turbulent flows and its relation to initial conditions and coherent structures. *Advances in Turbulence*, pages 39–73, 1989.

R. Giering, T. Kaminski, and T. Slawig. Generating efficient derivative code with TAF - Adjoint and tangent linear euler flow around an airfoil. *Future Generation Computer Systems*, 21(8):1345–1355, 2005.

M. B. Giles and N. A. Pierce. Adjoint equations in CFD: duality, boundary conditions and solution behavior. In *13th Computational Fluid Dynamics Conference Proceedings*, volume AIAA Paper 97-1850, Snowmass, CO, US, June 29-July 2 1997.

M. B. Giles and N. A. Pierce. An introduction to the adjoint approach to design. *Flow Turbulence and Combustion*, 65(3-4):393–415, 2000.

D. Greenblatt and I. J. Wygnanski. The control of flow separation by periodic excitation. *Progress in Aerospace Sciences*, 36(7):487–545, 2000.

A. Griewank and A. Walther. Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *Transaction on Mathematical Software*, 26(1), 2000.

A. Hay, D. Pelletier, and R. Di Caro. Verified predictions of shape sensitivities in wall-bounded turbulent flows by an adaptive finite-element method. *Journal of Computational Physics*, 228(12):4510–4531, 2009.

S. B. Hazra, V. Schulz, and J. Brezillon. Simultaneous pseudo-time stepping for 3D aerodynamic shape optimization. *Journal of Numerical Mathematics*, 16(2): 139–161, 2008.

A. Hilgers and B. J. Boersma. Optimization of turbulent jet mixing. *Fluid Dynamics Research*, 29(6):345–368, 2001.

M. Hinze and J. Sternberg. A-revolve: an adaptive memory-reduced procedure for calculating adjoints; with an application to computing adjoints of the instationary Navier–Stokes system. *Optimization Methods Software*, 20(6):645–663, 2005.

M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich. *Optimization with PDE constraints*, volume 23 of *Mathematical modelling: theory and applications*. Springer Science + Business Media B.V., New York, 2009.

A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(3), 1988.

A. Jameson and J. C. Vassberg. Computational fluid dynamics for aerodynamic design: Its current and future impact. *AIAA paper*, 0538, 2001.

A. Jameson, L. Martinelli, and N. A. Pierce. Optimum aerodynamic design using the Navier-Stokes equations. *Theoretical and Computational Fluid Dynamics*, 10(1-4):213–237, 1998.

L. Jenn-Long. Intelligent genetic algorithm and its application to aerodynamic optimization of airplanes. *AIAA Journal*, 43(3):530–538, 2005.

J. Jeong and F. Hussain. On the identification of a vortex. *Journal of Fluid Mechanics*, 285:69–94, 1995.

S. Kern. *Bioinspired optimization algorithms for the design of anguilliform swimmers*. PhD thesis, Eidgenössische Technische Hochschule ETH, 2007.

N. Kroll, N.R. Gauger, J. Brezillon, R. Dwight, A. Fazzolari, D. Vollmer, K. Becker, H. Barnewitz, V. Schulz, and S. Hazra. Flow simulation and shape optimization for aircraft design. *Journal of Computational and Applied Mathematics*, 203(2): 397–411, 2007.

S. Laizet, S. Lardeau, and E. Lamballais. Direct numerical simulation of a mixing layer downstream a thick splitter plate. *Physics of Fluids*, 22(1):15, 2010.

R. Lardat and M. Leschziner. A Navier–Stokes solver for LES on parallel computers. Technical report, UMIST, 1998.

D. G. Luenberger. *Optimization by Vector Space Methods*. John Wiley & Sons, inc, New York, 1969.

D. G. Luenberger. *Linear and nonlinear programming*. Boston, Kluwer Academic Publishers, 2nd edition, 2003.

R. W. Metcalfe, S. A. Orszag, M. E. Brachet, S. Menon, and J. J. Riley. Secondary instability of a temporally growing mixing layer. *Journal of Fluid Mechanics*, 184:207, 1987.

J. Meyers and P. Sagaut. Evaluation of Smagorinsky variants in Large-Eddy simulations of wall-resolved plane channel flows. *Physics of Fluids*, 19:095105, 2007.

Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag Berlin, Heidelberg, New York, 3rd edition, 1996.

A. Michalke. On the inviscid instability of the hyperbolic-tangent velocity profile. *Journal of Fluid Mechanics*, 19(4):543–556, 1964.

P. A. Monkewitz and P. Huerre. Influence of the velocity ratio on the spatial instability of mixing layers. *Physics of Fluids*, 25(7):1137–1143, 1982.

R. D. Moser and M. M. Rogers. The 3-Dimensional evolution of a plane mixing layer - pairing and transition to turbulence. *Journal of Fluid Mechanics*, 247: 275–320, 1993.

R. D. Moser, M. M. Rogers, and D. W. Ewing. Self-similarity of time-evolving plane wakes. *Journal of Fluid Mechanics*, 367:255–289, 1998.

J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, New York, 2 nd edition, 2006.

K. J. Nygaard and A. Glezer. Core instability of the spanwise vortices in a plane mixing layer. *Physics of Fluids a-Fluid Dynamics*, 2(3):461–464, 1990.

K. J. Nygaard and A. Glezer. Evolution of streamwise vortices and generation of small-scale motion in a plane mixing layer. *Journal of Fluid Mechanics*, 231: 257–301, 1991.

R. T. Pierrehumbert and S. E. Widnall. The two-dimensional and 3-dimensional instabilities of a spatially periodic shear-layer. *Journal of Fluid Mechanics*, 114 (Jan):59–82, 1982.

O. Pironneau. On optimum design in fluid mechanics. *Journal of Fluid Mechanics*, 64(1):97–110, 1974.

S.B. Pope. *Turbulent flows*. Cambridge University Press, 1 st edition, 2000.

W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in FORTRAN 77: The art of scientific computing*. Cambridge University press, 2nd edition, 1996.

B. Protas, T. R. Bewley, and G. Hagen. A computational framework for the regularization of adjoint analysis in multiscale PDE systems. *Journal of Computational Physics*, 195(1):49–89, 2004.

M. M. Rogers and R. D. Moser. The 3-Dimensional evolution of a plane mixing layer - the Kelvin-Helmholtz rollup. *Journal of Fluid Mechanics*, 243:183–226, 1992.

M. M. Rogers and R. D. Moser. Direct simulation of a self-similar turbulent mixing layer. *Physics of Fluids*, 6(2):903, 1994.

M. M. Rogers, R. D. Moser, and J. C. Buell. A direct comparison of spatially and temporally evolving mixing layers. *Bulletin of the American Physical Society*, 1990.

P. J. Schmid and D. S. Henningson. *Stability and transition in shear flows*, volume 142 of *Applied Mathematical Science*. Springer-Verlag New York, Inc, 2001.

J. A. Templeton, M. Wang, and P. Moin. An efficient wall model for large-eddy simulation based on optimal control theory. *Physics of Fluids*, 18(2), 2006.

B. Vandereycken and S. Vandewalle. A Riemannian optimization approach for computing low-rank solutions of Lyapunov equations. *SIAM J. Matrix Anal. Appl.*, 31(5):2553–2579, 2010.

R. W. C. P Verstappen and A. E. P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187(1):343–368, 2003.

T. Verstraete, Z. Alsalihi, and R. A. Van den Braembussche. Multidisciplinary optimization of a radial compressor for microgas turbine applications. *Journal of Turbomachinery-Transactions of the ASME*, 132(3):7, 2010.

B. Vreman, B. Geurts, and H. Kuerten. Large-eddy simulation of the turbulent mixing layer. *Journal of Fluid Mechanics*, 339:357–390, 1997.

M. J. Wei and J. B. Freund. A noise-controlled free shear flow. *Journal of Fluid Mechanics*, 546:123–152, 2006.

C. D. Winant and F. K. Browand. Vortex pairing: The mechanism of turbulent mixing layer growth at moderate Reynolds number. *Journal of Fluid Mechanics*, 63:237, 1974.

I. Wygnanski and H. E. Fiedler. 2-Dimensional mixing region. *Journal of Fluid Mechanics*, 41:327–361, 1970.

Curriculum vitae

Sara Delport was born in Leuven (Belgium) in 1983. She graduated as Master in Mechanical-Electrical Engineering with specialisation Mechanics-Aerospace Engineering at the Katholieke Universiteit Leuven in 2006. Her Master thesis was on the influence of the collector on the flow in a micro heat exchanger. Within the framework of her Master she worked as an intern at SONACA S.A. in Gosselies (Charleroi), Belgium in August 2005.

In August 2006 she started her Ph.D at the Katholieke Universiteit Leuven. The Ph.D was supervised by Prof. dr. ir. Tine Baelmans and Prof. dr. ir. Johan Meyers. The research was funded by a Ph.D grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). In April-June 2010 she was a research visitor at the department of Aeronautics, Imperial College London. The visit was funded by the FWO-Vlaanderen.

List of publications

S. Delport, M. Baelmans, and J. Meyers. Constrained optimization of turbulent mixing-layer evolution. *Journal of Turbulence*, 10(18):26, 2009.

S. Delport, M. Baelmans, and J. Meyers. Optimal control of dissipation in a turbulent shear flow. *Physics of Fluids*, submitted, 2010.

T. Stevens, F. Rogiers, S. Delport, P. Vleugels, J. Peirs and M. Baelmans. Collector pressure losses in micro heat exchangers. In *International Conference on Thermal Issues in Emerging Technologies - Theory and Applications, ThETA 1*, Cairo - Egypt, 2006.

S. Delport, M. Baelmans, and J. Meyers. Optimization of turbulent mixing restricted by linear and nonlinear constraints. In *ERCOFTAC WORKSHOP Direct and Large-Eddy Simulation 7 (DLES7)*, Trieste - Italy, 2008.

S. Delport, M. Baelmans, and J. Meyers. Optimization of long-term mixing in a turbulent mixing layer. In *5th AIAA Flow Control Conference*, Chicago - IL,US, 2010.

S. Delport, M. Baelmans, and J. Meyers. The effect of noise on optimal perturbations for turbulent mixing. In *ERCOFTAC WORKSHOP Direct and Large-Eddy Simulation 8 (DLES8)*, Eindhoven - The Netherlands, 2010.